МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН
Некоммерческое акционерное общество
«АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ»
Кафедра систем информационной безопасности

«ДОПУЩЕН К ЗАЩИТЕ»
Зав. кафедрой к.п.н., доцент Р. Ш. Бердибаев
_____ «_____» _____2019 г.
(подпись)

ДИПЛОМНЫЙ ПРОЕКТ

На тему: Быстродействующее устройство приведение чисел со сдвигом на 3 разряда промежуточных остатков
Специальность: 5В100200 - «Системы информационной безопасности»
Выполнил Нургалиев Толеген Маратулы Группа СИБ-15-2
Научный руководитель Тынымбаев Сахыбай Тнейбаевич

Консультант:
по экономической части:
к.т.н., профессор Аренбаева Ж.Г.
(ученая степень, звание, Ф.И.О)
(подпись) «10» мая 2019 г.

по безопасности жизнедеятельности:
д.т.н., Ст. преподаватель Бембасаров Ш.Ш.
(ученая степень, звание, Ф.И.О)
(подпись) «15» мая 2019 г.

по применению вычислительной техники:
к.т.н., профессор Тынымбаев С.Т.
(ученая степень, звание, Ф.И.О)
(подпись) «3» мая 2019 г.

Нормоконтролер:
Ст. преподаватель Аскарова Н.Б.
(ученая степень, звание, Ф.И.О)
(подпись) «5» июня 2019 г.

Рецензент:
_____
(ученая степень, звание, Ф.И.О)
_____ «_____» _____2019 г.
(подпись)

Алматы 2019

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН
Некоммерческое акционерное общество
«АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ»

Институт систем управления и информационных технологий

Кафедра систем информационной безопасности

Специальность 5В100200 - «Системы информационной безопасности»

ЗАДАНИЕ
на выполнение дипломного проекта

Студенту Нургалиеву Толегену Маратулы

Тема проекта Быстродействующее устройство приведение чисел со сдвигом на 3 разряда промежуточных остатков

Утверждена приказом по университету №124 от «26» апреля 2018 г.

Срок сдачи законченного проекта «6» июня 2019 г.

Исходные данные к проекту (требуемые параметры результатов исследования (проектирования) и исходные данные объекта): проект подразумевает разработку быстродействующего устройства , позволяющего уменьшить аппаратную нагрузку на систему ассиметричных криптоалгоритмов .В качестве исходных данных быстродейсвующего устройства будут принимать следующие данные: тактовый импульс и двоичный код . Помимо обширного функционала устройства , необходимо реализовать формирователь частичного остатка . В качестве формирователя частичного остатка использоваться: на основе схем сумматора и на основе схем сравнения .Также указывается на проблему арифметическими вычисления над числами повышенной разрядностью(возведению степень по модулю)

Перечень вопросов, подлежащих разработке в дипломном проекте, или краткое содержание дипломного проекта: дипломный проект включает в себя 6 глав, разделенных на подглавы, каждая из которых освещает определенную тематику, используемую при разработке быстродействующего устройства .

В первой главе дипломного проекта представлена общая информация по шифрования : типы шифрования ,RSA , плюсы и минусы, надежность и аппаратного шифрования.

Во второй главе дипломного проекта представлена методы приведения чисел по модулю: представлены различные способы и примерная реализация на примере различных схем .

В третей главе подробно описывается функционал каждого устройства по отдельности.

В четвертой главе был реализован алгоритм нашего устройства

В пятой главе приводится технико-экономическое обоснование, показывающее актуальность предприятия с финансовой точки зрения.

В шестой главе рассматриваются раздел БЖД подробно рассматривается и вычисляется искусственное и естественное освещения .

Перечень графического материала (с точным указанием обязательных чертежей):

1 схема устройства приведение чисел со сдвигом на 3 разряда ;

2 схема ФЧО на основе сумматоров ;

3 схема ФЧО на основе схем сравнений ;

4 схема на устройства для формирователя остатка по модулю

Основная рекомендуемая литература:

1 Аманжолова К. Б., Алибаева С. А. Экономика предприятии телекоммуникации: Учебное пособие. - Алматы: АИЭС, 2003

2 Орлов Г.Г., Булыгин В.И., Виноградов Д.В., Иващенко П.Ф., Коптев Д.В., Пчелинцев В.А., Ройтман В.М., Шапошников В.Н. Инженерные решения по охране труда в строительстве, – М.: Просвещение, 1985

3 Бекишева А.И. Методические указания к выполнению экономической части дипломной работы для бакалавров специальности 5В0703 - Информационные системы – Алматы: АУЭС, 2013.

4 СНиП РК 2.04-05-2002 «Естественное и искусственное освещение».

Конструкции по проекту с указанием относящихся к ним разделов проекта

Раздел	Консультант	Сроки	Подпись
6 глава	Алибаева М.Г.	04.03 – 10.05.19	
5 глава	Бекбасаров Ш.Ш.	04.03-30.05.19	
4 глава	Тлеулесов С.Т.	04.03-30.05.19	
1 глава.	Тлеулесов С.Т.	04.03-30.05.19	
2 глава	Тлеулесов С.Т.	04.03-30.05.19	
3 глава	Тлеулесов С.Т.	04.03-30.05.19	

График
подготовки дипломного проекта

Наименование разделов, перечень разрабатываемых вопросов	Сроки представления научному руководителю	Примечание
Изучение ассиметричной криптографии.	07.04.2019	
Изучение методов приведения чисел.	10.04.2019	
Изучение ФРО на основе сумматоров	25.04.2019	
Разработка ФРО на основ схем-сравнения.	30.04.2019	
Разработка блок-управляющего устройства	12.05.2019	
Исправление ошибок.	15.05.2019	
Экономическая часть	18.05.2019	
Безопасность жизнедеятельности.	20.05.2019	

Дата выдачи задания «05» марта 2019 г.

Заведующий кафедрой _____ (_____)
(Подпись) (Ф.И.О)

Научный руководитель проекта _____ (Тлепшибаев С.Т.)
(Подпись) (Ф.И.О)

Задание принял к исполнению студента _____ (Нурланов Т.К.)
(Подпись) (Ф.И.О)

АННОТАЦИЯ

В дипломной работе подробно разбираются широкая область применения аппаратное шифрования в современном мире, структура работы компонентов аппаратного значения. Также уделяется внимание история аппаратного шифрования история возникновения, современная конкурентоспособность. Была реализован алгоритм моего устройства. Был реализован алгоритм нашего устройства на языках программирования. В следующих главах рассматривается экономическая эффективность в целом, а также освещенность здания.

АҢДАТПА

Дипломдық жұмыста заманауи әлемде аппараттық шифрлаудың кең қолдану аймағы, аппараттық маңызы бар компоненттердің жұмыс құрылымы егжей-тегжейлі талданады. Сондай-ақ аппараттық шифрлау тарихы пайда болу тарихы, қазіргі заманғы бәсекеге қабілеттілік назар аударылады. Менің құрылғымның алгоритмі іске асырылды. Біздің құрылғымыздың алгоритмі бағдарламалау тілінде жүзеге асырылды. Келесі тарауларда жалпы экономикалық тиімділік, сондай-ақ ғимараттың жарықтануы қарастырылады.

ANNOTATION

The thesis examines in detail the wide scope of hardware encryption in the modern world, the structure of the hardware components . Attention is also paid to the history of hardware encryption history of origin, modern competitiveness. The algorithm of my device was implemented. The algorithm of our device was implemented in programming languages .The following chapters examine the overall cost–effectiveness as well as the illumination of the building.

Содержание

Введение

Криптозащита один из способов защитить информацию. Криптоалгоритмы делятся на симметричные и ассиметричные шифрования. Симметричные шифрования используют один ключ. Ассиметричное шифрование – шифрование которая используют открытой и закрытой ключ. Открытый ключ используется по уязвимого каналу а для расшифровки сообщение используют закрытый ключ. Криптография с открытым ключом очень тесно связана с идеей односторонних функций, односторонние функция – функция после которой практически невозможно получить исходное данные. Ассиметричные системы шифрования теорически обладают высокой степени защиты. Но недостатком их является слабая скорость обработки информации. Во многих криптосистемах ассиметричного шифрования используется возведения в степень по модулю многоразрядности. Для ускорения возведения в степень существует различные решения в схемотехнике. Имеются аппаратные решения для ускорения приведения по модулю которые требует много аппаратных затрат. То же время существует простые решение для приведения по модулю которые характеризуется низкой скоростью но приемуществом данных решение является минимальные затраты. Эти решения используют обыденной способ получения остатка путем вычитания модулю из проводимого числа и получаемых остатков. В абсолютно любом ассиметричном шифрование используется операция чисел по модулю .

Аппаратное шифрование имеет ряд существенных преимуществ перед программным шифрованием, одним из которых является более высокое быстродействие, чем программная реализация. Аппаратная реализация криптоалгоритма гарантирует его целостность, а шифрование и хранение ключей осуществляется в самой плате шифратора, а не в оперативной памяти компьютера. Это очень важно для обеспечения защищенной реализации самого алгоритма, что также является важным преимуществом аппаратной. Наиболее эффективным методом борьбы с такими угрозами является хранение и передача особо важных данных в зашифрованном виде. Криптографические методы защиты информации могут быть реализованы программно, аппаратно, программно-аппаратно. Аппаратное шифрование имеет ряд существенных преимуществ перед программным шифрованием, одним из которых является более высокое быстродействие, чем программная реализация. Но аппаратное реализация является более дорогим нежели программно. Базовой операцией для криптосистем с открытым ключом является возведение целых чисел в степень по модулю P (axmodP). Эта процедура реализуется применением операций «умножение», «возведение в квадрат» и «приведение по модулю». Поэтому для повышения производительности криптосистем с открытым ключом требуется разработка алгоритмов, ускоряющих выполнения этих операций.

Для ускорения операций умножения и возведения в квадрат можно использовать массивы двоичных сумматоров, матрицу Брауна, дерево Уоллеса, счетчики Дадда, систолические и ведические умножители и другие способы. Однако эти умножители и квадраторы эффективны при вычислении «малоразрядных» операндов, которые нашли широкое применение при построении множительных блоков для компьютеров различных классов. Таким образом, тема дипломной работы является весьма актуальной в наше время Для достижения поставленной цели необходимо решить следующие задачи:

- изучить теоретические основы компонентов шифрования, их классификацию;
- рассчитать экономическую эффективность;
- изучить роль освещения в здании.

Информационной базой данной дипломной работы является учебно-методологическая литература отечественных и зарубежных авторов, интернет-ресурсы.

1 Теоретическая часть

1.1 История создания и область применения

История криптография имеет очень долгую историю. Историки говорят что она история ее около 4 тысяча лет. На начало времен (приблизительно с 3-го тысячелетия до н. э.) характеризуется господством примитивных шифров (основной принцип замена букв другими буквами или символами). Второй период с 9 века – до начала 20 века ознаменовался введением в обиход алфавитных шифров. Суть его проста очередная буква заменяется символом из собственного алфавита, причем каждый следующий алфавит получается из предыдущего путём сдвига на одну букву. Третий период (с начала и до середины 20 века) характеризуется внедрением электромеханических устройств в работу криптографии. При этом продолжалось использование алфавитных шифров. Четвёртый период – с середины до 70-х годов XX века – переход к математической криптографии. Современный период развития криптографии (с конца 1970-х годов по настоящее время) отличается зарождением и развитием нового направления – криптография с открытым ключом. Её появление интересно не только новыми возможностями, но и сравнительно широким распространением криптографии для использования частными лицами. Правовое регулирование использования криптографии частными лицами в разных странах сильно различается во всем мире –от разрешения до полного запрета. Современная криптография это наука между информатикой и математикой. Сейчас в 21 веке почти во всех сферах используется криптоалгоритмы такие как коммерция, общение в социальных сетях и т.д.

Особенности систем шифрования напрямую зависят от специфики объекта, на котором они используется . Сфера применения шифрования широка – такие меры обеспечения безопасности используются во многих областях человеческой деятельности. Список задач, которые система шифрования на предприятии поможет решить наиболее эффективно и с минимальными затратами, обширен. Это:

- организация охраны данных , принадлежащей предприятию;
- защита секретных данных ;
- предохранение от НСД.

Возможности контроля расширятся с применением программного обеспечения и технических усовершенствований системы шифрования. Криптозащита в банках. Системы шифрования в банковской системе позволяет защитить информацию о вкладчиках и защитить репутацию банка. С их помощью решаются такие важные задачи, как:

- обеспечение максимальной безопасности;
- контроль данных.

Система шифрования в социальных сетях. Шифрования, интегрированное в социальных сетях, могут обеспечить защиту данных от нежелательного взлома, рассылки информации.

Стоить отметить, что в социальных сетях в мессенджерах используется сквозное шифрования. Сквозное шифрования передача данных, в котором только люди, участвующие в общении, имеют доступ к сообщениям. Так использование сквозного шифрования не позволяет получить доступ к криптографическим ключам со стороны третьих лиц. Для обмена ключами могут быть применены ассиметричные и симметричные системы шифрования. Разработчики говорят то что с помощью сквозным шифрованием информация может быть известно только людям говорящим с друг с другом. Система шифрования существенно повышает безопасность данных людей и имущества. Дополнительное программное обеспечение позволит полностью контролировать рабочие процессы даже дистанционно, через Интернет.

В свою очередь аппаратные средства шифрования имею 3 вида разновидности:

- шифровальные модули (которую работу делают всю сами);
- блоки шифрования в каналах связи;
- шифровальные платы для установки для ПК.

Шифрование возможно осуществить программно, аппаратно и программно-аппаратно. Аппаратное шифрование имеет ряд существенных преимуществ перед программным шифрованием – аппаратные средства шифрования обладают большей скоростью (аппаратная реализация любого алгоритма, в том числе и криптографического, обеспечивает более высокое быстродействие, чем программная реализация); – аппаратуру шифрования легче физически защитить от проникновения извне, чем программу; – аппаратуру шифрования проще установить. Поэтому большинство средств криптографической защиты данных реализовано в виде специализированных аппаратных устройств. Эти устройства встраиваются в линию связи и осуществляют шифрование всей передаваемой по ней информации. Преобладание аппаратного шифрования над программным шифрованием обусловлено не только указанными выше причинами, перечень достоинств аппаратных шифраторов значительно шире:

- аппаратная реализация криптоалгоритма гарантирует его целостность;
- шифрование и хранение ключей осуществляются в самой плате шифратора, а не в оперативной памяти компьютера;
- аппаратный датчик случайных чисел создает действительно случайные числа для формирования надежных ключей шифрования и электронной цифровой подписи;
- на базе аппаратных шифраторов можно создавать системы защиты информации от несанкционированного доступа и разграничения доступа к компьютеру;

− применение специализированного шифрапроцессора для выполнения криптографических преобразований разгружает центральный процессор компьютера;

− возможна также установка на одном компьютере нескольких аппаратных шифраторов, что еще более повышает скорость обработки информации;

− использование шин в архитектуре шифрапроцессора исключает угрозу снятия ключевой информации по возникающим в ходе криптографических преобразований колебаниям электромагнитного излучения в цепях "земля – питание" микросхемы.

1.2 Классификация криптоалгоритмов

Шифрованиие делятся на два вида симметричные и ассиметричные .Коротко говоря симметричные использует один ключ ассиметричные два ключа. Симметричные криптосистемы делятся на два вида :

− блочные системы шифрования суть его состоит в том то что информация разбиваем на блоки одинаковый длины ,следующий шаг каждый блок отдельно шифруем единственным ключом в заключение весь шифроблок складывается;

− потоковые шифры – где информация переводится биты с использованием гаммирования. В потоковых криптосистеме используется ГПСЧ. ГПСЧ выдает определенную числовую последовательность последняя накладывается на шифруемую информацию с применением операции XOR. Операция XOR это прежде всего строгая дизъюнкция. Особенностью симметричных шифрованний является скорость и простота реализации.

В асимметричных методах шифрования используются два ключа. Один ключ является закрытым и известным только получателю. Его используют для расшифрования. Второй из ключей является открытым, он может быть общедоступным по сети и опубликован вместе с адресом пользователя. Его используют для выполнения шифрования. Понятно, что ключ расшифрования нельзя определить из ключа зашифрования. Самым популярным из асимметричных является метод RSA (Райвест, Шамир, Адлеман), основанный на операциях с большими (скажем, 100-значными) простыми числами и их произведениями. Которое подробно мы рассмотрим в следующий главе.

1.3 RSA

Алгоритм RSA является первым алгоритмом шифрования с открытым ключом. Название системы RSA происходит от первых букв фамилий ее авторов – Р.Ривест,А. Шамир и Л.Адлеман. Система базируется на следующих фактах:

Для шифрования используется операция возведение в степень по модулю.Для дешифрования используется функция Эйлера. Особенность данной крипто системы является трудность разложение на множители больших чисел. В RSA используется открытый и закрытый ключ ,открытый

передается по уязвимого каналу ,а закрытые используется для получения исходного значния. Открытый и закрытый ключ математически с друг другом связаны что сообщение зашифрованные одним ключом можно дешифровать только вторым ключом из этой пары. Количество натуральных чисел меньших n и взаимно простых с n называется функцией Эйлера и обозначается ф(n).

Надёжность шифрования обеспечивается тем, что третьему лицу очень трудно вычислить закрытый ключ по открытому. Оба ключа вычисляются из одной пары простых чисел (p и q). То есть ключи связаны между собой. Но установить эту связь очень сложно. Основной сложностью является разделение модуля n на простые множителя p и q. Если число является произведением двух очень больших простых чисел, то его очень трудно разложить на множители.Есть и недостатки криптоалгоритма RSA например избегается применения чисел менее 200 десятичных разрядов. Криптосистема RSA реализуется аппаратно и программно. Для аппаратной реализации RSA были сделаны специальные процессоры. Эти процессоры, реализованные на сверхбольших интегральных схемах (СБИС), позволяют выполнять операции RSA, связанные с возведением больших чисел в очень большую степень по модулю P, за относительно короткое время. Одна из самых быстрых аппаратных реализаций RSA с модулем 512 бит на сверхбольшой интегральной схеме имеет быстродействие 64 Кбит/с. Тем не менее, аппаратная реализация RSA выполняет операции шифрования и дешифрования примерно в 1000 раз медленнее, чем аппаратная реализация DES – симметричного криптоалгоритма. Программная же часть реализации данного криптоалгоритма Такой существенный разрыв в быстродействии возникает из–за того, что в RSA используется возведение очень больших (многоразрядных) чисел в очень большую степень по модулю P. Алгоритм RSA активно реализуется как в виде самостоятельных криптографических продуктов, так и в качестве встроенных средств в приложениях. Например, для защиты баз данных в серверах используются встроенные механизмы шифрования, которые предусматривают использование RSA. Определим базовые операции над числами, которые используются в асимметричных криптоалгоритмах шифрования. Возведение чисел в степень по модулю P (a x mod p) реализуется через использование таких операций как умножение, возведение в квадрат и приведение по модулю. И одним из подходов для повышения производительности криптосистем с открытым ключом, является ускорение выполнения этих операций. Самой сложной из них является операция приведения по модулю, так как она представляет собой получение остатка от деления числа на модуль P, а операция деления – самая сложная из арифметических операций. И эта операция повторяется многократно, так как вместо многократного умножения и затем деления очень большого числа (a x) на модуль, для ускорения возведения в степень по модулю, используется многошаговое последовательное умножение с приведением по модулю на каждом шаге каждый раз нового произведения. При этом также понижается разрядность перемножаемых чисел и, соответственно, разрядность

произведения, подлежащего перемножению. Например, если нужно вычислить a 16 mod p, то вместо выполнения пятнадцати перемножений и одного приведения по модулю очень большого числа (a*a*a*a*a*a*a*a*a*a*a*a*a*a*a*a) выполняют четыре возведения в квадрат, используя после каждого возведения в квадрат приведение по модулю: a 16 mod p = (((a 2) 2) 2) 2 mod p = (((a 2 mod p) 2 mod p) 2 mod p) 2 mod p. Это позволяет уменьшить разрядность операндов и ускорить возведение чисел в степень по модулю P. И чем длиннее число, тем заметнее ускорение.

2 Практическая часть

2.1 Методы приведения чисел по модулю

Возведение в степень по модулю – одна из действий над числами – выполняемая по модулю. Примеряется в криптографии. Есть много методов формирования остатков при делении на модуль. Если использовать двоичного представления целых положительных чисел можно выделить три способа формирования остатков по произвольному модулю Р. В первом способе кратные модули Р*i (i=1,3…k) формируются в разных блоках, затем они с использованием К сумматоров одновременно (параллельно) вычитаются из приведенного числа А. Наименьший положительный остаток Ci=A-P*i является результатом . Такой способ формирования остатков характеризуется большими аппаратными затратами – при больших соотношениях приводимого числа А и модуля Р сложность схемы резко возрастает. Этот способ приемлем по аппаратным затратам лишь при малых значениях А и Р .Второй способ основан на последовательном формировании остатков (ri) разрядных весов двоичного числа (2i) от деления на модуль Р с дальнейшим суммированием по модулю Р тех остатков, для которых коэффициенты Ai соответствующих весов равны единице и реализуется по формуле: A $mod\ P$ = (∑ (2 $imod\ P$)Ai)$mod\ P$. k-1 0 . Так как для двоичной системы счисления коэффициенты Ai (i=0÷k-1), принимают только два значения (0 и 1), то суммируя заранее вычисленные частичные остатки по модулю от числа 2i (i=0,1,…k-1) для тех i, для которых коэффициенты Ai=1, получают остаток по модулю Р от числа А. Так как для двоичный системы счисления кооэффицент a(i=0,,K-1)принимается только буллевые значение сложением заранее вычисленные остатки по модулю Р от числа А. Частичный остаток 20 для любого модулю (Р>2) всегда равен 1.ЧО от 21 в два раза превышает ЧО 2 0 таким образом частичный остаток 2i в два раза превышает 2i-1.Аксиомой этого метода является вычисление в умножении на два частичного остатка 2i-1 и приведение числа по модулю А. Операция умножения на 2 может быть реализована сдвигом влево. Операция приведение по модулю Р для чисел не превышающих 2Р-1 реализуется по другому. Если число не превышает значение 2Р-1 из него вычитается модуль Р,а результат является остатком.

В третьем способе приведения числа по модулю использован принцип машинного алгоритма двоичного деления со сдвигом остатков влево. Модуль Р последовательно вычитается, начиная со старших разрядов числа А. На каждом шаге вычитания формируется очередной частичный остаток. Для вычисления следующего частичного остатка предыдущий остаток сдвигается влево на один разряд и к младшему разряду частичного остатка присоединяется следующий разряд числа А. И из полученного числа вычитается модуль. Частичные остатки формируются в формирователях частичных остатков ФЧО. Совокупность всех формирователей частичных остатков образует матричную схему приведения числа А по модулю Р.

Недостатком первого способа является необходимость иметь в устройстве деления сумматора и регистра делителя двойной длины. Второй способ позволяет обойтись узлами одинарной длины, поэтому первый способ деления мы рассматривать не будем.

Если числа A и P положительны, то частное Q и остаток R будут положительными. Последовательный алгоритм деления сдвигает число A и вычитает число P от A до тех пор, пока не будет найден остаток R, удовлетворяющий условию $0<R<n$. При этом не исключено получение отрицательного остатка после вычитания.

Деление чисел может быть произведен с восстановлением, либо без восстановления остатков. При этом в состав делителя входит схема «исключающее ИЛИ», обеспечивающая передачу делителя на вход сумматора либо в прямом, либо в инверсном коде в зависимости от знака очередного остатка. Это приводит к усложнению схемы делительного устройства.

Деление можно сделать двумя способами :

− с неподвижным делитилем и сдвигаемым вправо делителем;

− с неподвижным делителем и сдвигаемым влево делителем.

При аппаратной реализации приведения по модулю могут быть использованы самые различные подходы, которые приводят к большому разнообразию структур устройств получения остатка от деления на модуль. Эти структуры представлены в различных публикациях, но систематизация и анализ их отсутствует. Анализ структур и принципов функционирования различных устройств приведения по модулю позволил выявить их характерные признаки:

− последовательное или параллельное выполнение операций возведения в квадрат и получения остатков от деления на модуль;

− однотактность или многотактность работы устройства;

− наличие или отсутствие схемы управления (управляющего автомата) операцией приведения по модулю;

− использование определенной системы счисления. С учетом этих характеризующих признаков все устройства приведения по модулю могут быть разбиты на классы. Ниже предлагается классификация устройств приведения по модулю на основе указанных выше критериев.

Классификация по степени параллельности процессов умножения и приведения произведения по модулю:

− параллельные – приведение по модулю осуществляется в процессе умножения, параллельно. После получения каждого частичного произведения каждый раз выполняется его приведение по модулю и в дальнейшем для продолжения умножения используется не частичное произведение, а его остаток;

− последовательные – приведение по модулю осуществляется после получения произведения, последовательно. Выполняется умножение на a или возведение a в квадрат, только потом находят его остаток от деления на модуль.

Классификация по количеству тактов, необходимых для получения остатка в устройстве приведения по модулю:

− многотактные устройства, в которых остаток определяется путем многократного вычитания из исходного приводимого числа, а впоследствии из полученных положительных остатков, модуля, по которому осуществляется приведение. И здесь возможны два варианта: все вычитания реализуются на одних и тех же узлах, которые многократно циклически участвуют в процессе получения каждого остатка (циклическая организация); – вычитания реализуются на аппаратном конвейере (конвейерная организация), каждая схема которого используется только один раз. Каждый остаток формируется на своем уровне конвейера, количество которых определяется максимальным количеством положительных остатков;

− однотактные устройства, в которых параллельно выполняются вычитания из приводимого числа модуля P и чисел, кратных модулю (2P, 3P, 4P. При этом получают множество остатков, результатом является наименьший положительный остаток.

Классификация по наличию управляющего автомата (УА) в устройстве приведения по модулю:

− комплексное устройство – представляется в виде совокупности операционного и управляющего автоматов (ОА и УА). УА вырабатывает управляющие сигналы и управляет процессом приведения по модулю, а все операции выполняются в ОА. Операционный автомат, в свою очередь, посылает осведомительные сигналы в управляющий автомат, которые служат для управляющего автомата ориентиром при выработке очередного управляющего сигнала. Это типичный случай классического операционного устройства (ОУ), при синтезе которого применимы известные методы синтеза цифровых автоматов, в том числе и микропрограммных автоматов (МПА). Здесь возможны следующие варианты: – управляющий автомат может быть построен в виде схемы – УА с жесткой логикой; – управляющий автомат может быть построен на основе принципа программного управления – УА с программируемой логикой;

− автономное устройство – не выделяется управляющая часть, всѐ реализовано в виде единой схемы, управляющие сигналы формируются в результате выполнения операций.

Класификация по системе счисления, используемой в устройстве приведения по модулю:

− двоичная система счисления;

− двоично–десятичная система счисления.

Вспомогательные системы счисления с основанием 2^h , где h целое число и $h>=2$. Переход к вспомогательной системе счисления осуществляется условно из двоичной системы счисления путем разбиения двоичного числа на диады (h=2, 2^h =4), на триады (h=3, 2^h =8), на тетрады (h=4, 2^h =16) и т.д.

2.2 Формирование остатков по модулю P

Способ формирования остатка основан на последовательном формировании остатков (r_i) разрядных весов двоичного числа (2^i) от деления по модулю P с дальнейшим суммированием по модулю P тех остатков, для которых коэффициенты a_i соответствующих весов равны единице.

Тогда формула для вычисления остатка r_A от числа A по модулю P имеет следующий вид:

$$r_A = AmodP[\int_{i=0}^{k-1}(2^i modP)a_i]modP \quad (2.1)$$

где 2^i – вес i-го разряда числа A(i=0/k-1), a_i – коэффицент i-го разряда числа A.

Так как для двоичной системы счисления коэффиенты a_i (I = 0, … K-1), принимают только два значения (0 или 1), суммируя заранее вычисленные остатки по модулю P от числа 2^i (I = 0, … K-1), для тех i, для которых коэффицент a_i=1, а получают остаток по модулю P от числа A. Частичный остаток от 2^0 для любого модуля (P≥2) всегда равен единице. Частичный остаток от 2^1 в два раза повышает остаток от 2^0 и т.д., т.е. частичный остаток 2^i в два раза превышает частичный остаток от 2^{i-1}. Таким образом, вычисление частичного остатка от 2^i заключается в умножении на два частичного остатка от 2^{i-1} и приведении результата по модулю A. Операция умножения на два может быть реализована сдвигом всех разрядов умножаемого числа на один разряд влево. Так как для двоичной системы счисления коэффиенты a_i (I = 0, … K-1), принимают только два значения (0 или 1), суммируя заранее вычисленные остатки по модулю P от числа 2^i (I = 0, … K-1), для тех i, для которых коэффицент a_i=1, а получают остаток по модулю P от числа A. Частичный остаток от 2^0 для любого модуля (P≥2) всегда равен единице. Частичный остаток от 2^1 в два раза повышает остаток от 2^0 и т.д., т.е. частичный остаток 2^i в два раза превышает частичный остаток от 2^{i-1}. Таким образом, вычисление частичного остатка от 2^i заключается в умножении на два частичного остатка от 2^{i-1} и приведении результата по модулю A. Операция умножения на два может быть реализована сдвигом всех разрядов умножаемого числа на один разряд влево.

Операция приведения по модулю P для чисел, не превышающих величину 2P-1, реализуется следующим образом. Если число не превышает величину P, то оно остается без изменения, если же число лежит в интервале от P до 2P-1, то из него вычитается модуль P, а результат является остатком.

На рисунке 2.1 представлена функциональная схема устройства для формирования остатков, на рисунке 2.2 – функциональная схема формирователя частичных остатков, на рисунке 2.3 – функциональная схема сумматора по модулю (СММ).

Устройство формирования остатков по модулю (Рисунок 2.1) состоит из

n–1 последовательно соединенных ФЧО, схем $И_0 \div И_{k-1}$ и К-1 сумматоров по модулю. Разряды числа А $a_0 \div a_{n-1}$ из регистра А подаются не соответствующие схемы $И_0 \div И_{k-1}$, где логически умножаются со значениями частичного остатка $r_0 \div r_{k-1}$.

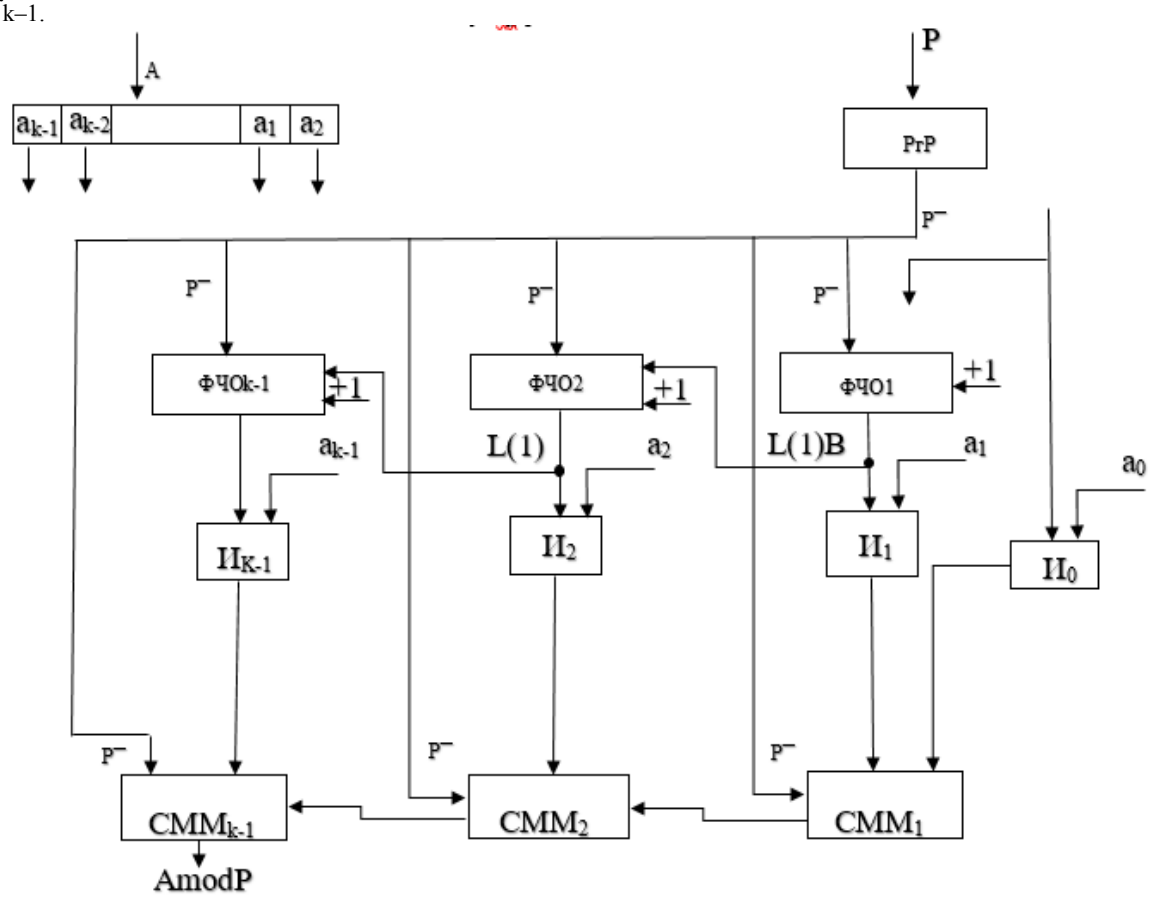

Рисунок 2.1 – Функциональная схема устройства для формирователя остатка по модулю.

Инверсное значение модуля (P^-) подается на входы $ФЧО_1 \div ФЧО_{k-1}$ $СММ_1 \div СММ_{k-1}$. На первом шаге вычисления остатка значение 2^0= (частичный остаток r_0) подается на первый вход схемы $И_0$, а на второй вход a_0 – при a_0=1 на выходе $И_0$ формируется промежуточный остаток R_0, который на втором шаг вычисления остатка подается на вход $СММ_1$. Одновременно на втором шаге вычисления частичный остаток r_1 с выхода $ФЧО_2$, со сдвигом на один разряд влево подается на вход $ФЧО_1$ также и без сдвига подается н первый вход схемы $И_2$ на второй вход которого подается значение разрядного коэффициента a_2. При a_2=1 с выхода $И_2$ r_1 подается на вход сумматора п модулю $СММ_2$ и на его выходе формируются промежуточный остаток R_1.

Для вычисления остатка от числа А по модулю Р достаточно в формуле (4) просуммировать частичные остатки по модулю Р от чисел $2^{2i}(2a_{2i}+ 2_{2i})$.

Способ вычисления частичных остатков от 2^{2i} по модулю Р состоит в следующем. Вычисление частичного остатка от 2^{2i} заключается в умножении на четыре частичного остатка от $2^{2(i-1)}$ и приведение результата по модулю Р. Операция умножения на четыре может быть реализовала сдвигом всех

разрядов числа на два разряда влево. При этом операция приведения по модулю реализ
Если число лежит в интервале от 3p до 4p-1, то из нею вычисляется утроенный модуль – 3P.

Способ умножения частичного остатка от 2^{2i} по модулю P на число $(2_{2i}+a_{2i})$ состоит в следующем. Частичный остаток от 2^{2i} по модулю P, умноженный на $2a_{2i}$ складывается с частичным остатком от 2^{2i} по модулю P, умноженн
Если число лежит в интервале от 2p до 3p-1, то из него вычитывается удвоенный моду
умножитель по модулю (УММ). Схема устройства умножения по модулю два (СММ) была приведена на рисунке 2.3.

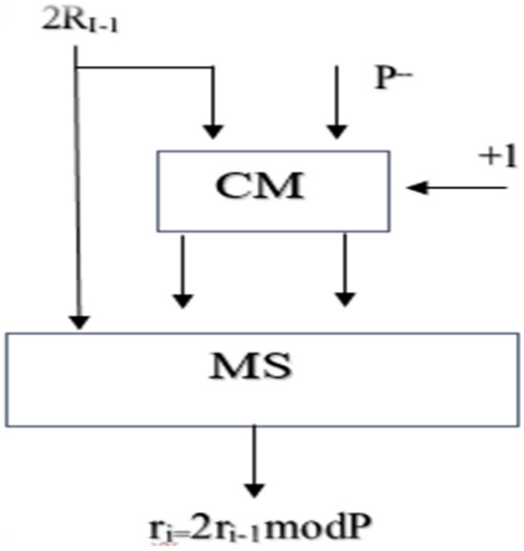

Рисунок 2.2 – Функциональная схема формирователя частичного остатка

Устройство для формирования остатка содержит К схем И, К/2 ФЧО, УММ. ФЧО между собой соединены последовательно, причем на вход $ФЧО_0$ подан код единицы, разряд которого сдвинут на два разряда влево. Выходы разрядов предыдущего ФЧО подаются на входы последующего (L2) ФЧО со сдвигом на два в сторону старших (L2). На входы каждого ФЧО подаются значения модуля P, удвоенное 2p и утроенное P значение 3p формируется на сумматоре СМ0.

На информационные входы УММ подаются коды с выходов ФЧО и значения пары разрядов $2a_0a_0$, $2a_1a_1$, $2a_2a_2$, ... $2a_{n-1}a_{n-1}$ на информационные входы CMmodP подаются коды с выходов УММ и с выходов предыдущего CMmodP, причем на нулевой сумматор по модулю P два младших разрядов a_i и a_0 кода числа.

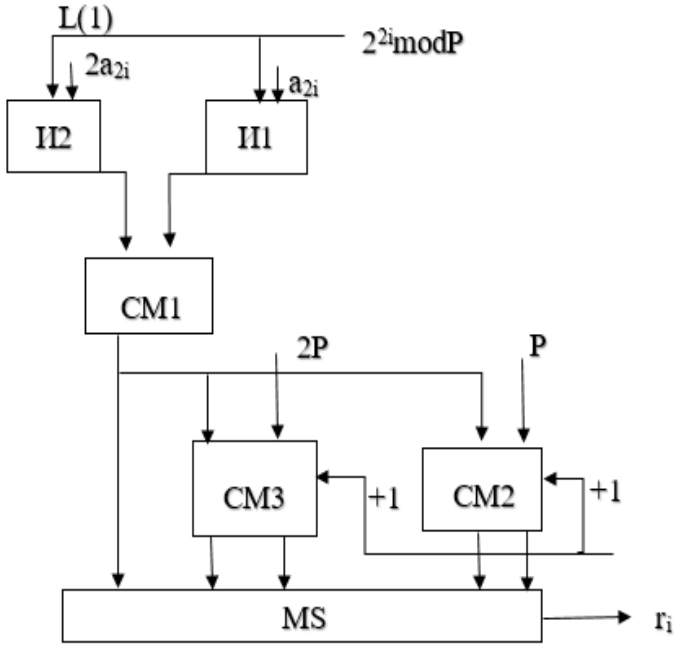

Рисунок 2.3 – Функциональная схема сумматора по модулю (СММ)

Каждый ФЧО (Рисунок 2.5) содержит три сумматора (СМ1÷СМ3) и мультиплексор (MS). На первые информационные выходы трех сумматоров подается код с выхода ФЧО. На вход СМ3 подается инверсный код утроенного модуля 3P. Удвоенный код модуля 2P подается на вход СМ2, а инверсный код модуля P подается на вход сумматора СМ1.[3]

Устройство формирования остатков по модулю (рисунок 1) состоит из n–1 последовательно соединенных ФЧО, схем $И_0 \div И_{k-1}$ и К–1 сумматоров по модулю. Разряды числа А $a_0 \div a_{n-1}$ из регистра А подаются не соответствующие схемы $И_0 \div И_{k-1}$, где логически умножаются со значениями частичного остатка $r_0 \div r_{k-1}$.

Мультиплексор MS коммутирует свои входы с выходами определенного сумматора или информационною выхода, куда подается значение разрядных весов 2^{2i} в зависимости от соотношения 2^{2i}с величинами 3P, 2P, P.

Каждый УММ (Рисунок 2.6) содержит двух схем И1 и И2, который управляются разрядами кода числа А, три сумматора СМ1, СМ2, СМ3 1 мультиплексора MS. На входы СМ3 подается удвоенный инверсный код, модуля 2P и на вход СМ2 подается P. Код с выходов СМ1 подается на вход, MS и СМ3, и СМ2.

На первом этапе вычисления остатка на вход $УММ_0$ подается $r_0=2^0=1$ на выходе $УММ_0$ формируется промежуточный остаток $R_0=2°(2a_0+a_0)$. Кром этого значение r_0 сдвинутое в сторону старших двух разрядов т.е. $4r_0=$ подается на вход $ФЧО_1$.

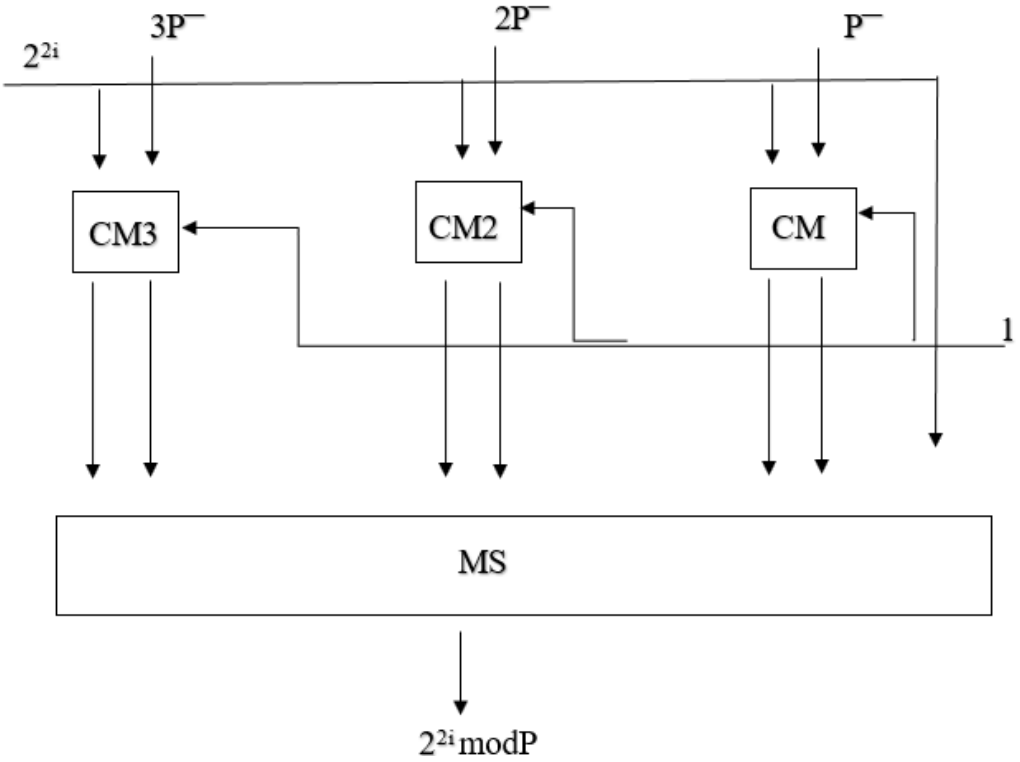

Рисунок 2.4 – Структура ФЧО

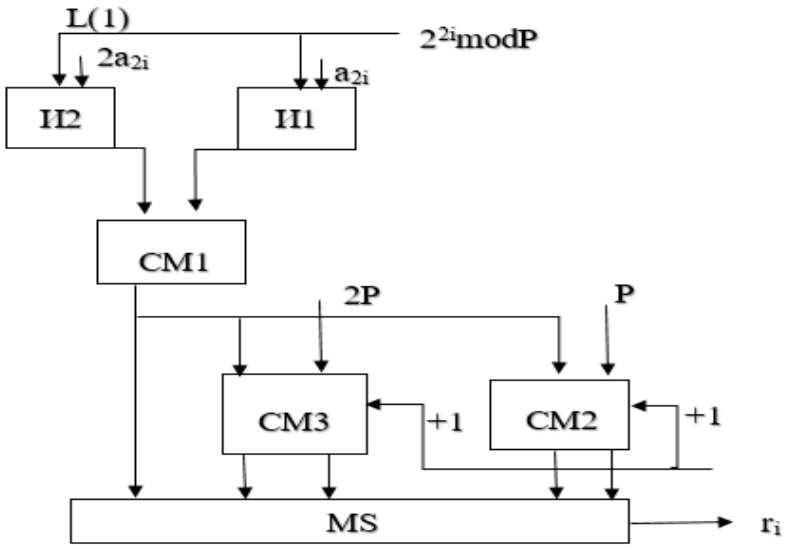

Рисунок 2.5 – Структура умножителя по модулю (УММ)

На втором этапе на выходе $ФЧО_1$ формируется частичный остаток r_i который подается на вход $УММ_1$ где r_1 умножается на $2a_3$ a_2. Затем он суммируются по модулю P и формируется r_i, который подается на вход $СММ_1$, где вычисляется промежуточный остаток $R_i=(R_0+r_i)modP$, кроме этого на втором шаге значение r_i со сдвигом на 2p влево (L2) подается на информационный вход $ФЧО_2$, а на входы $СММ_2$ подается значение Ri и $СММ_2$

На третьем этапе на выходе $ФЧО_2$ формируется частичный остаток значение которого подается на вход $УММ_2$ где r_2’ умножается на значение

разрядов a_4 и $2a_5$. Затем они суммируются по модулю P формируется, частичный остаток r_2, далее частичный остаток r_2 подается на вход $СММ_2$. В $СММ_2$ r_2 суммируется с R_1 по модулю и формируется промежуточный остаток $R_2^{=}(R_1+r_2)modP$. Аналогично определяются другие r и R. После n/3 этапов на выходе $УММ_{n/2}$ формироваться результат.

Из рис. 4 видно, что устройство формирования остатка по модулю P со сдвигом приводимого числа влево на два разряда остаток формируется за K/2 шагов и в каждом шаге частичный остаток r, проходит через $ФЧО_{i,}$ $УММ_i$, $СММ_i$, то время формирования остатка определяется:

$$T_{фо} = 1/2(T_{ФЧО} + T_{УММ} + T_{СММ}) = 1/2 = (T_{см} + 2T_{СМ} + 2Tсм) = 2{,}5T_{СМ} \quad (2.2)$$

Количество сумматоров:

$$N_{СМ} = 1/2\ (N_{ФЧО} + N_{УММ} + N_{СММ}) = 1/2\ (5N_{СМ} + 3\ N_{СМ} + 2\ N_{СМ}3) = ½*8\ N_{СМ} = 4\ N_{СМ} \quad (2.3)$$

Операция приведения по модулю P для чисел, не превышающих величину 2P-1, реализуется следующим образом. Если число не превышает величину P, то оно остается без изменения, если же число лежит в интервале от P до 2P-1, то из него вычитается модуль P, а результат является остатком.

Способ формирования остатка путем параллельного вычитания чисел кратных модулю A. Однотактовые устройство приведение чисел характеризуется большими аппаратными затратами.В таких устройствах параллельно разные блоки формируется кратные модули PxI(где i=2,3,…k).Затем модуль P и сформированные кратные модули 2p,3p…,kp с использованием к сумматоров и одновреммено вычитается число A .Наименьший положительный остаток R=A+P+1 является результатом.Для формирование кратныз 2p , 3p,…kp требуется K-1 сумматоров. С увеличением числа $div = \frac{A}{P}$ резко увеличивается число сумматоров для вычисления значения остатка и формирователей кратных i×p.Если взять div= 6 для формирования кратныз p,2p….6p потребуется 6 сумматоров и 3 формирователя кратных $3\bar{P}$ $4\bar{P}$ $5\bar{P}$.

Количество сумматоров:

$$Ncm=1{,}5(Nkp+N)=1{,}5KNсм \quad (2.4)$$

При делении числа A на число P нас интересует не частное Q ,а остаток R.Для данных делимого A и делителя P частное Q и остаток R вычисляется так чтобы выполнялось соотношене.

Недостатком первого случая является нужно иметь устройство деление сумматор и регистр делителя двойной длины.Второй же способ позволяет обойтись узлами одинарной длины.Если число A P положительны то часное Q

то остаток R будут положительными.Последовательный алгоритм сдвигает до тех пор пока не будет найден остаток R.Подходяющию условию $0 \le R < n$.Однако может получиться отрицательный остаток имеено поэтому отличается деление с восстановлением остатка и безвосстановления остатка.

2.3 Деление с неподвижным делимым и сдвигаем право

В ЭВМ операция деление чисел с помощью соответсвующих алгоритмов сводится к операциям вычитания и сдвига.Реализовать деление можно двумя способами как уже оговаривалось выше.Рассмотрим случай на примере.

$$Z=X/Y \qquad (2.5)$$

X– делимое ,представляемое словом(2n–1)

Y– делитель

Z – частное содержащими n-1 цифровых разрядов.

Для обобщения оговариваем что числа которые делятся ,представляется в коде. Так как Z частное ((n–1) разрядное число то диапазон от 0 до 2^{n-1} .Это возможно только при $(|X| - |Y|)<0$ $|Y| = |Y| * 2^{(n-1)}$. Для получения $(|X| - |Y|)$ следует отнять из |X| делитель |Y|,выровняв их так что бы разряд Y был под n-м разрядом делимого. Этого возможно получить сдвинув делитель Y относительно делимого X на n-1 разрядов влево.

Если результат вычитания $(|X| - |Y|)$ (это вычитание называют пробным) больше 0 ,то $Z > 2^{n-1}$ и деление невозможно если меньше 0 ,то возможно выполнить деление. Недостатком такого способа является двойная длина СМ и его регистров в АЛУ. Рассмотрим пример:

X = -38 Y = 7.Представим делимое и делитель в прямом коде старший разряд – знаковый который в плюсе равен 1 а минусе 0.$X_{пр}$=10100110, Y = 0111,тогда модуль делимого и делителя в их старших разрядов заносятся в нули. Частное Z должно быть представлено в коде с 4 двоичными разряда .Выполняем деление 10100110/0111. Справилами деления очередной цифрой частного является 1, если после вычитания из остатка делителя получается положительный результат, и 0 если результат отрицателен.

2.4 Деление с неподвижным делителем и сдвигаемым влево делителем

Способ позволяет распалогать АЛУ с n разрядными регистрами и сумматорами. Данный способ имеет две разновидности:

− деление с неподвижным с восстановлением остатка;

− деление с неподвижным делителем без восстановление остатка.

Деление с восстановлением остатка–можно разложить след способом:

− исходное значение частичного остатка полагается полагается на старшим разрядом делимого;

− чо удвадваивается путем сдвига на 1 разряд влево;

− из двинутого ЧО вычитается делитель ;

− модуль положителен если число равно 1 и 0 если отрицателен. В случае 0 остаток восстановляется до того значение которое было до вычитания.

Деление с без восстановление остатка можно разложить следующим способом:

− исходное значение ЧО полагается равным старшим разрядом делимого;

− чо удваивается путем 1 сдвиг влево.

Недостатком описанных выше алгоритмов – необходимость выполнения на отдельных шагах дополнительных операций сложения для восстановления остатка. Эти способы увеличивает время деления которое может меняться в зависимости сочетания кодов операндов. В силу указанных причин реальные делители строятся на основе без восстановления остатка.

3 Быстродействующие устройства приведения чисел по модулю со сдвигом на 3 разряда

Рассмотрим алгоритм деления чисел с со сдвигом на 3 разрядаФункциональная схема устройства приведения чисел по модулю приведение чисел со сдвигом на 3 разряда промежуточных остатков

Рассмотрим функционирование этой схемы. Сигналом ПУСК посредством блока схем И1 в РгА принимается 3/n шагов .для формированию частиных остатков потребовалось 7 двоичных сумматоров что приводит облегчению работы.Наше устройство состоит из 2n сдвигающего регистра на 3 разряда влево PrA ,где хранича 2n разрядное приводимое число А из блока формирования модулей 7Р формирователя частичных остатков схем ИЛИ Сумматоров и состоит из множества регистров а также счетчика СЧТИ

Регистр PrA состоит 3/n разряда 3 сдвигающего влево и сохранимого числа А а n разрядного PrP – разрядный Pr3P – для хранение модуля Р и утроенного модулю Р. На входы сумматоров СМ3 СМ2 подается инверсное значение Р и инверсных выходов регистров Pr3P Pr2P со сдвигом на влево .На выходе ФЧО значение r определяется путем анализа соотноешний значений остатака ri–1 умноженное на восемь 8ri–1 со значением кодов один разряд а на сумматоров СМ1 значение инверсное Р и ргегистр PrP без сдвига.Частичный остаток ri–1 сдивнутый на 3 разряда влево формируется ФЧО .Если же 8ri–1 <P то формируется значения ЗН = 1 при этом П= 0 тогда сигналом ЗН =1 значения 8ri–1 – Р используется значения И ,ИЛИ .При этом 8ri–1. При этом есть необходимость выполнения операции сложения делителя Р с частичным остатком, что приводит к добавлению. Схем «исключающее ИЛИ» из тракта «делитель-сумматор».

Функциональная схема устройства приведения чисел со сдвигом на 3 разряда промежуточных остатков. Рассмотрим функционирование этой схемы. Сигналом ПУСК число и модуль принимается в регистр Pr и в блок формирование модулей 7Р и 7$\bar{Р}$ тактовых импульсов (СчТИ) принимается двоичный код числа шагов приведения К в счетчик тактовых импульсов (СчТИ). Записовается двоичный код от числа n K =log n/3 который сдвигается на 3 разряда формируя 8ri–1 который передается в ФЧО которого остается частиный отстаток r который передается PrA1 и отпраляется И6 .ФЧО состоит из сумматора и схем сравнения и мультиплексора. С поступлением ТИ2 показание СЧти уменьшается еще на единицу. Он выходит конец операций СчТИ = 0 который выходит.

Теперь давайте рассмотрим схему ФЧО который содержит 8 схем сравнения сумматор и схемы сравнения И схемы ИЛИ в состав ФЧО добавлен блок логических схем И9, формирующий значение 8ri–1 при Зн3=Зн2=Зн1=1.

После К шагов на выходе ФЧО формируется частичный остаток r шагов. Как видно в состав ФЧО добавлены схемы И и ИЛИ при различых значениях

Зн количество двоичных сумматоров при этом определяется по формуле Q=3(K-1)Ncm T=время суммирование, K разряда сумматоров и модуля P. Инверсное значение P подается на входы ФЧО. На входы сумматора и левые входы схем сравнения сдвинутый в право на 3 разряда с присоеднинеем. Коммутация один из P÷6P осуществляется схемами И6 ÷И12 в зависимости от сравнения 8ri-1. Со значительными модулями P 7p.Например если при 8ri-1<P на выходе 1 в СС-1 формируется сигнал 1 который через И0 и И3 выдает значение 8ri-1. На выход при соотношении P< 8ri-1.<2P на выходе 2СС-1 (8ri-1 > P) и на выходе 1 СС-2 (8ri-1<2P) выбабатывается сигнал 1 и сигнал 1 с выхода системы И1 схемой И7 коммутирует значение P .На правые входы сумматора СМ и одновременно +1 подается на вход младшего разряда сумматора ,Передовая обратный код P в дополнительный и на выходе формируется R1 путем выполнения операций r=8ri-1+ P+1 .Аналогично при соотношение 5P<8ri-1<6p на выходе 2 СС -5 (8ri-1>5P) и на выходе 1 СС-6 (8ri-1<6P) вырабатываются единичные сигналы,которые подаются на выходы схемы И5 коммутирют значение 5P на правые входы СМ и этот сигнал подаетя на вход младшего разряда Сумматора и на выходе сумаатора формируется остаток R путем выполнения операции Ri–1 = 8ri-1 + 5P +1 .

При 7P<8ri-1 на выходе 2 СС-7 вырабатывается единичный сигнал который посредством схемы И13 коммутирует значение 7P на правые входы сумматора 1 сигнал с выхода 2 СС-7 также через схему ИЛИ 2 на вход младшего сумматора, где выполнения Ri=8ri-1+7P +1

Таким образом включая в состав одного ФЧО и сдвигая регистр в каждом такте на 3 разряда три раза уменьшитьь число записей и сдвига частичных остатков. В состав устройство входят 2n+2 разрядный регистр сдвига на два разряда влево РгА, где хранится приводимое число А, регистр РгР для хранения n-разрядного модуля Р, сумматор СМ1 для вычисления обратного кода устроенного значения модуля (путем суммирования с). Значения 3p формируется путем инвертирования выходных разрядов СМ1 на блоке инверторов НЕ1. В состав устройство также входят схемы сравнения СС-1, СС-2 и СС-3, где значение 8ri-1 сравнивается со значениями модулей 3p, 2p и p, схема И1, схема ИЛИ3, блоки схем И2÷И13, блоки схем ИЛИ1 и ИЛИ2, линий задержки л.з.1÷л.з.3, триггер Т, вычитающий счетчик тактовых импульсов СчТИ. На схеме СС-1 сравниваются коды 4ri-1 и p. Если при этом 8ri-1 < p, то на ее выходе 1 формируется единичный сигнал, на ее выходе 2 формируется сигнал «0». И наоборот, если 8ri-1 ≥ p, то на выходе 2 этой схемы формируется сигнал «1», а на выходе 1 сигнал «0». Схемой СС-2 сравниваются значение 4ri-1 со значением модуля 2p. На выходе 1 этой схемы установится сигнал «1», если 4ri-1 < 2p, при этом на выходе 2 установится «0». Если 8ri-1 ≥ 2p, то на выходе 1 установится сигнал «0», а на выходе 2 сигнал «1». Схемой СС-3 сравниваются коды 4ri-1 и 3p. На выходе 1 этой схемы формируется сигнал «1», если 4ri-1 < 3p, при этом на выходе 2 установится «0». Если при этом 4ri-1 ≥ 3p, то на выходе 1 формируется сигнал «0», а на выходе 2 установится сигнал «1». После поступления в схему n/2-го

тактового импульса в СчТИ установится код нуля и вырабатывается сигнал «Конец операций», который подается на нулевой вход триггера Т и блокирует прохождение следующего тактового импульса на выход схемы И1. Последним тактовым импульсом вычисляется последний остаток rn-1, который запоминается в старших n разрядах регистра РгА, что является результатом вычисления. Задержанным на л.з.2 сигналом «Конец операций» результат из РгА выводится блоком схем И12 на выход (Таблица 3.1).

Таблица 3.1 – Определение для вычисления остатка ri по результатам сравнения 8ri-1 с со значением P÷7P на выходах схем сравнения СС-1 ÷СС-7

п/н	Соотношения	Выходы схем сравнений	Выполняемые операции
1	8ri-1<P	СС-1-1(8ri-1<P)	Передача ri=8ri-1 в регистр остатка
2	P≤8ri-1<2P	СС-1-2 (8ri-1≥P) СС-2-1 (8ri-1<2P)	ri=8ri-1-P
3	2P≤8ri-1<3P	СС -2-2(8ri-1≥P) СС-3-1(8ri–1<2P)	ri=8ri-1-2P
4	3P≤8ri-1<4P	СС-3-2(8ri-1≥3P) СС-4-1(8ri-1<4P)	ri=8ri-1-3P
5	4P≤8ri-1<5P	СС -4-2(8ri–1≥4P) СС -5-1(8ri-1<5P)	ri=8ri-1-4P
6	5P≤8ri-1<6P	СС-5-2(8ri-1≥5P) СС-6-1(8ri-1<6P)	ri=8ri-1-5P
7	6P≤8ri-1<7P	СС-6-2(8ri-1≥6P) СС-7-1(8ri-1<7P)	ri=8ri-1-6P
8	7P≤8ri-1	СС -7-2(8ri-1≥7P)	ri=8ri-1-7P

Согласно этой таблице при 8ri-1 < p значение 4ri-1 схемами И6, ИЛИ 4 и ИЛИ1 без изменения записываются в РгА. При соотношениях p ≤ 8ri-1 < 2p на выходе схемы И7 формируется единичный сигнал, который одновременно подается на вход ИЛИ3 и блока схем И9, на вторые входы блока И9 подаются разряды, которые через блок схем ИЛИ2 подаются на правые входы сумматора СМ2. На правые входы подводятся коды значения 4ri-1, а на вход младшего разряда СМ2 схемой ИЛИ3 подается сигнал «+1» и в сумматоре выполняется операция 8ri = ri-1. Результат через блок схем ИЛИ4 и ИЛИ1 передается в старшие разряды регистра РгА. При выполнения условий 8ri-1 ≥ 2p и 8ri-1 < 3p на выходе схемы И8 формируется единичный сигнал, который подается на вход схемы ИЛИ3 и схемы блока И10, на вторые информационные входы которого подаются разряды модуля. Разряды модуля через блок схем ИЛИ2 поступают на правые входы сумматора СМ2, а на вход младшего разряда подается код «+1» и в сумматоре выполняется операция ri = 4ri-1 + +1. Результат вычисления через блок схем ИЛИ4 и ИЛИ1 передается в

регистр РгА. При соотношений 8ri-1 ≥ 3p со второго выхода схемы сравнения СС-3 единичный сигнал подается на вход схемы ИЛИ3 и на управляющий вход блока схем И11. На информационные входы этой схемы подаются разряды модуля p, умноженного на три с выходов сумматора СМ1. Коды 3 через блок схем ИЛИ2 передается на правые входы сумматора СМ2,н а левые входы которого подаются разряды кода 8ri-1. При этом в сумматоре выполняется операция ri = 8ri-1. Результат операций через блок схем ИЛИ4 и ИЛИ1 записываются в старшие разряды регистра РгА.

3.1 Структура устройства

Счетчик команд неотлиемлимый объект устройства управление ВМ. Ск – выполняет важнейшую функцию.Счетчик команд заносит адрес ячейки основной памяти где хранится команда которая должна выполнится первой.В рассматриваемой ВМ любая команда занимает одну ячейку поэтому содержимое увеличивается на 1 что обеспечивает подачу сигнала .По завершению текущий команды адрес следующий команды всегда берется из СК. Для изменения естественного порядка вычислений надо занести в СК точки перехода. Регистр адреса – использует для хранения испольнительных адресов операндов.

Сумматор – устройства логический операционный узел, выполняющий арифметическое сложение двоичных, троичных или nn–ичных кодов. Может складывать два (бинарный), три (тернарный) или nn чисел (nn-арный). Помимо сложения выполняются и другие операции: учёт знаков чисел, выравнивание порядков слагаемых.

Схема сравнения – цифровые компараторы относятся к арифметическим устройствам. Цифровые компараторы выполняют сравнение двух чисел, заданных в двоичном (двоично–десятичном) коде. В зависимости от схемного исполнения компараторы могут определять равенство A=B (A и B) – независимые числа с равным количеством разрядов) либо вид неравенства: A<B или A>B. Устройство приведения по модулю состоит из блока регистров (БлРг), формирователя частичного остатка (ФЧО), блока синхронизации (БлС).

БлРг состоит из регистра А (РгА), который имеет цепи сдвига влево на один разряд. Разрядность РгА–2n. РгА служит для хранения, делимого А, который приводится по модулю Р. РгР служит для хранения делителя – модуля Р. Разрядность которого – n.

Блок частичного остатка (БЧО) состоит из сумматора (СМ) и из п разрядного блока «исключающего ИЛИ», работающего в режиме управляемого инвертора.

Блок синхронизаций (БлС) состоит из линий задержки Л3.1, триггера Т схем И2 и И1, вычитающего счетчика тактовых импульсов СчТИ и триггера знака ($Т_{зн}$), где запоминается знак остатка после очередного вычитания Р из А.

Старшие (2n÷n) разряда РгА через схему И5 связаны с левыми входами сумматора СМ, а на левые входы СМ подаются разряды РгР прямом или

обратном коде в зависимости знака очередного остатка. Значение знака остатка подается вход младшего разряда сумматора.

Выходы сумматора через схему ИЛИ связаны со старшими разрядами РгА, где запоминаются частичные остатки, подлежащие к сдвигу в сторону старшего разряда. После завершения операций значение остатка из регистра РгА[2_{n-1}-n] выдается по сигналу «Конец операции» схема И7.

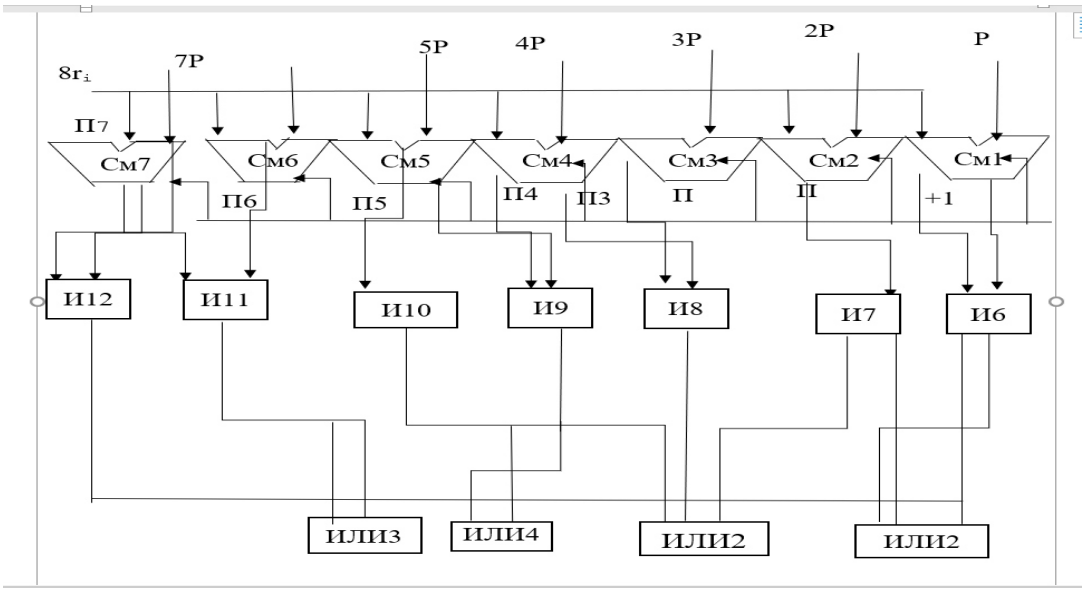

Рисунок 3.1 – Формирователь частичных остатков и разрядов частного (ФЧО и РЧ-3-1)

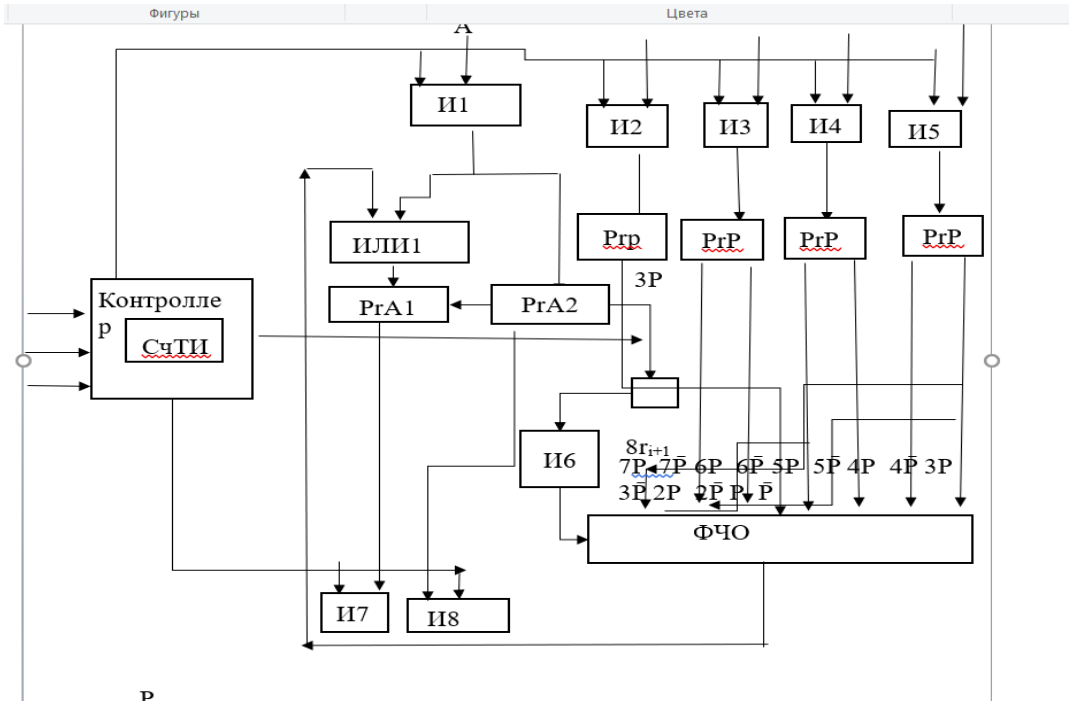

Рисунок 3.2 – Устройства приведение чисел со сдвигом на 3 разрядов ЧО

На входы ФЧО поступают утроенные значения модуля 3P и 3$\bar{P}$ из

соответствующего регистра и значения модуля Р в прямом и обратном кодах – Р и $\bar{P}$. Значения $2\bar{P}$ и 2Р формируются путем сдвига влево значений соответственно $\bar{P}$ и Р влево на один разряд. Кроме того, на вход ФЧО подается значение предыдущего остатка со сдвигом на два разряда влево т.е. 8ri-1 и его инверсное значение. На выходе сумматора в результате одного из трех сложений 8ri-1 +$3\bar{P}$+1, 8ri–1 +$2\bar{P}$+1 или 4r+$\bar{P}$+1 формируется частичный остаток – ri.

Умноженный на четыре предыдущий частичный остаток 8ri-1 подается на первые входы сумматора СМ и на первые входы схем сравнения СС-1, СС-2, СС-3. На вторые входы СС-1 подаются значение Р и его инверсное значение $\bar{P}$. что упрощает структуру СС-1. Аналогично на вторые входы СС-2 подаются 2Р и $2\bar{P}$. На входы СС-3 подаются значения 3Р и $3\bar{P}$. На схеме СС-1 сравниваются коды 8ri-1 с кодом модуля Р. Если при этом 8ri-1≥Р, то на выходе 2 этой схемы формируется сигнал 1. который подается на вход схемы И1. Если 8ri-1<Р, то на ее выходе 1 формируется единичный сигнал. Схемой СС-2 сравниваются коды 8ri-1 и удвоенного модуля 2Р. На выходе 1 этой схемы установится сигнал «1», если 8ri-1<2Р. при этом на выходе 2 установится «О». Если 8ri-1≥2Р. то на выходе 1 установится сигнал «0», а на выходе 2 сигнал «1». Схемой СС-3 сравниваются коды 8ri-1 и 3Р. На выходе 1 это схемы формируется сигнал «1», если 8ri-1<3Р, при этом на выходе 2 установится «0». Если при этом 8ri-1≥3Р, то на выходе1 формируется сигнал «0», а на выходе 2 установится сигнал 1.

При соотношениях Р≤8ri-1<2Р на выходе схемы И1 формируется единичный сигнал, который одновременно подается на ИЛИ2 и блок схем ИЗ. На вторые входы ИЗ подаются разряды модуля в инверсном коде Р, которые через блок схемы ИЛИ1 подаются на правые входы сумматора СМ. На левые входы сумматора подается код 8ri-1 , а на вход младшего разряда сумматора схемой ИЛИ2 подаются сигнал +1 и в сумматоре выполняется операция ri=8ri-1+$\bar{P}$+1.

При соотношениях 8ri-1>2Р и 8ri-1<3Р на выходе схемы И2 формируется единичный сигнал, который подается на вход схемы ИЛИ2 и блока схем И4, на вторые информационные входы которого подаются разряды удвоенного модуля в инверсном коде $2\bar{P}$. Значение модуля $2\bar{P}$ через блок схемы ИЛИ1 поступают на правые входы сумматора СМ, а на младший разряд сумматора подается код +1 и в сумматоре выполняется операция ri=8ri-1+$2\bar{P}$+1.

При соотношениях 8ri-1≥3Р со второго выхода схемы сравнения СС-3 единичный сигнал подается на вход блока схем И5. На информационные входы этой схемы подаются разряды модуля $3\bar{P}$. Коды $3\bar{P}$ через блок схемы ИЛИ1 передаются на правые входы сумматора. При этом в сумматоре выполняется операция ri=8ri-1+$2\bar{P}$+1.

В устройствах приведения по модулю последовательного действия при условиях 8ri-1<Р в регистре остатков сохраняется предыдущий остаток. В матричных схемах при 8ri-1<Р значение 8ri-1 через схему И0 и блок

логической схемы ИЛИ3 передается следующему ФЧО.

Старшие разряды регистра РгА через блок схемы И4 связаны с ФЧО. Посредством И4 под управлением ТИ, поступающего из контроллера передается значение 8ri-1 из регистра РгА. На входы ФЧО подаются значения 3P, 3$\bar{P}$, 2P, 2$\bar{P}$ и P, $\bar{P}$. Выходы ФЧО через блок схемы ИЛИ очередной частичный остаток подается на входы РгА. На входы контроллера подается сигнал ПУСК, тактовые сигналы ТИ и число сдвигов n/2, необходимых для вычисления R=AmodP. Результат вычисления по сигналу «Конец операций» посредством блока схем И5 выдается на выход. Сначало идет использование регистра Pr5 Pr3 PrP оттуда выходит значение 6P 5P 4P 3P 2P P и идет использования инверсное значение P. прошлой схеме формирователь частичного остатка был основан на основе сумматора .Но на использовании сумматора является аппаратно затратным поэтому схема ФЧО будет использована на основе схем сравнения . зависимости от соотношения кода 8ri-1 и значений модулей 3p, 2p и p вырабатывается единичный сигнал либо на выходе 1 схемы СС-1, либо на выходе схемы И7, или И8, либо на выходе 2 схемы СС-3. По выработанному единичному сигналу выполняется операция вычисления ri согласно таблицы 1. Полученный частичный остаток ri блок схем ИЛИ4 и ИЛИ1 передается в регистр РгА, запоминаясь в старших разрядах регистра РгА. К этому моменту в схему поступает второй тактовый импульс ТИ2, который проходит через блок схем И1 и сдвигает РгА еще на два разряда влево, формируя значение 4ri. Одновременно ТИ2 поступает на вычитающий вход СчТИ и уменьшает его состояние на единицу. Значение 8ri с выхода блока И5 поступает на входы СМ2, СС–3, СС–2, СС–1 и сумматором СМ2 вычисляется промежуточный остаток ri, который передается в старшие разряды РгА.Также не стоит забывать то что инверсное значение подается в схемы И12 И11 И10 И9 И8 И7 подаются в схему ИЛИ1 оттуда модули 6P 5P 4P 3P 2P P подаются схему ИЛИ2 в конце подается значение ТИ оттуда в значение в в правые входы сумматора .Прибавляется значение +1 .После сдвига на 1 остается на линии задержки Л.ЗВ устройствах приведения по модулю последовательного действия при условиях 8ri–1<P в регистре остатков сохраняется предыдущий остаток. В матричных схемах при 8ri–1<P значение 8ri–1 через схему И0 и блок логической схемы ИЛИ3 передается следующему ФЧО. Схемой СС–3 сравниваются коды 8ri–1 и 3P. На выходе 1 это схемы формируется сигнал «1», если 8ri–1<3P, при этом на выходе 2 установится «0».

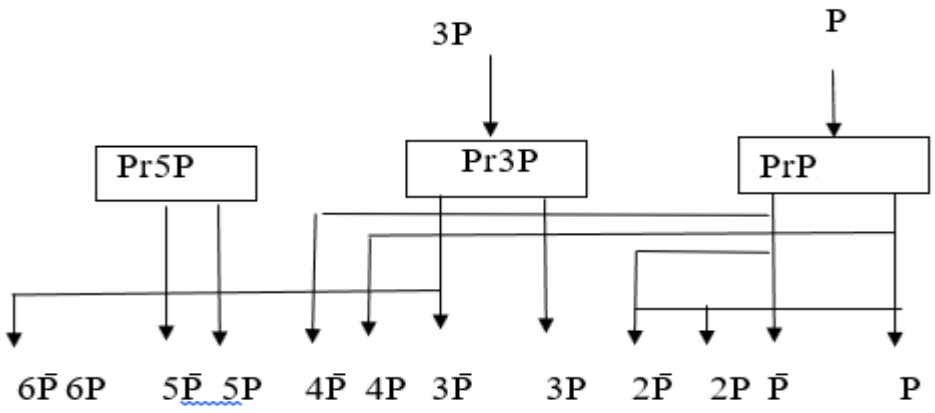

Рисунок 3.3 – Структура ФЧО

3.2 Матричная и конвейеризированная схемы приведения по модулю

При построении матричных схем приведения числа по модулю частичные остатки с выхода очередного $ФЧО_i$ передается со сдвигом на один разряд влево на входы следующего $ФЧО_{i+1}$. Функциональная схема формирователя остатка (ФЧО) приведена на рисунке 3.4

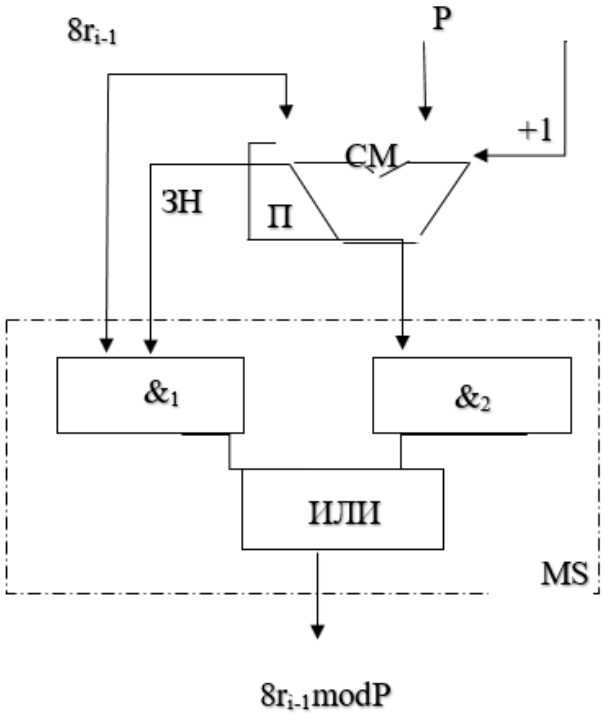

Рисунок 3.4 – Функциональная схема формирователя частичного остатка

Если удвоенный предыдущий остаток $8r_{i-1} \geq P$ то после окончания $8r_{i-1} - P$ сумматор вырабатывает перенос П=1 и при этом знак разницы ЗН=0, тогда разница $8r_{i-1}$-P с выходов сумматора СМ через схему $И_2$ передается на выход схемы ИЛИ. Если $8r_{i-1} < P$, то в знаковом разряде разницы $2r_{i-1}$-P формируется

ЗН=1 и при этом П=0, тогда сигналом ЗН-1 значение $2r_{i-1}$ через схему И1 и ИЛИ передается на выход. При этом $r_i = 8r_{i-1}$.

Теперь на основе вышерассмотренного ФЧО можно построить матричную схему устройства приведения числа А по модулю Р. На рис. 11 приведена структурная схема матричной схемы приведения по модулю для чисел А= a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0 и Р= P_3 P_2 P_1 P_0. Матричная схема работает следующим образом. По сигналу «Пуск» приводимое Число А принимается в регистр РгА, а разряды модуля Р принимается в регистр РгР. Инверсный код модуля Р и уровень +1 подается на входы всех формирователей остатков $ФЧО_0$ – $ФЧО_3$ значения частичного остатка r= a_7 a_6 a_5 a_4 подается на вход $ФЧО_0$ и одновременно значение $ФЧО_0$ подается значения $2r_0$ из которого вычитается значение Р путем сложения к rr_0 модуль Р в дополнительном коде и вычисляется частичный остаток $r_1 = 8r_0 + P+1$. После формирования частичного остатка и этот остаток удваивается путем сдвига на два разряда влево и пристыковывая к нему разряд a_2 подается на входы $ФЧО_3$ где формируется $r_2 = 8r_1 + P+ 1$. В следующие моменты времени на входах $ФЧО_2$, ФЧОз формируется частичные остатки r_3 и r_4. Значение r_4 является величина R= r_4=AmodP.

Устройства приведения числа А по модулю приведена структурная схема матричной схемы приведения по модулю для чисел А= a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0 и Р= P_3 P_2 P_1 P_0. Матричная схема работает следующим образом. По сигналу «Пуск» приводимое Число А принимается в регистр РгА, а разряды модуля Р принимается в регистр РгР. Инверсный код модуля Р и уровень +1 подается на входы всех формирователей остатков $ФЧО_0$ – $ФЧО_3$ значения частичного остатка r= a_7 a_6 a_5 a_4 подается на вход $ФЧО_0$ и одновременно значение $ФЧО_0$ подается значения $2r_0$ из которого вычитается значение Р путем сложения к rr_0 модуль Р в дополнительном коде и вычисляется частичный остаток $r_1 = 8r_0 + P+1$.

Счетчик команд неотлиемлимый объект устройства управление ВМ.Ск– выполняет важнейшую функцию.Счетчик команд заносит адрес ячейки основной памяти где хранится команда которая должна выполнится первой.В рассматриваемой ВМ любая команда занимает одну ячейку поэтому содержимое увеличивается на 1 что обеспечивает подачу сигнала .По завершению текущий команды адрес следующий команды всегда берется из СК. Для изменения естественного порядка вычислений надо занести в СК точки перехода.

Регистр адреса – использует для хранения испольнительных адресов операндов. 3Р со второго выхода схемы сравнения СС-3 единичный сигнал подается на вход блока схем И5. На информационные входы этой схемы подаются разряды модуля $3\bar{Р}$. Коды $3\bar{Р}$ через блок схемы ИЛИ1 передаются на правые входы сумматора. Количество двоичных сумматоров при этом определяется по формуле Q=3(K-1)Ncm T=время суммирование, K= разряда сумматоров и модуля Р.

Инверсное значение Р подается на входы ФЧО.На входы сумматора и

левые входы схем сравнения сдвинутый в право на 3 разряда с присоеднинеем. Коммутация один из P÷6P осуществляется схемами И6 ÷И12 в зависимости от сравнения 8ri-1. Со значительными модулями P 7p. Например если при 8ri-1.<P на выходе 1 в СС-1 формируется сигнал 1 который через И0 и И3 выдает значение 8ri-1. На выход .При соотношении P< 8ri-1<2P на выходе 2 СС -1 (8ri-1 > P) и на выходе 1 СС-2 (8ri-1<2P) выбабатывается сигнал 1 и сигнал 1 с выхода системы И1 схемой И7 коммутирует значение P. На правые входы сумматора СМ и одновременно +1 подается на вход младшего разряда сумматора ,Передовая обратный код P в допольнительный и на выходе формируется R1 путем выполнения операций r=8ri–1.+ P +1 .Аналогично при соотношение 5P<8ri-1<6p на выходе 2 СС-5 (8ri-1 >5P) и на выходе 1 СС-6 (8ri-1<6P) вырабатываются единичные сигналы,которые подаются на выходы схемы И5 коммутирют значение 5P на выходыформируется сигнал 1 который через И0 и И3 выдает значение 8ri-1. На выход ppи соотношении P< 8ri-1<2P на выходе 2 СС-1 (8ri-1> P) и на выходе 1 СС-2 (8ri-1<2P) выбабатывается сигнал 1 и сигнал 1 с выхода системы И1 схемой И7 коммутирует разряды.

Принимается только буллевые значение сложением заранее вычисленные остатки по модулю P от числа A. Частичный остаток 20 для любого модулю (P>2) всегда равен 1. ЧО от 21 в два раза превышает ЧО 2 0 таким образом частичный остаток 2i в два раза превышает 2i-1. Аксиомой этого метода является вычисление в умножении на два частичного остатка 2i–1 и приведение числа по модулю A. Операция умножения на 2 может быть реализована сдвигом влево. Операция приведение по модулю P для чисел не превышающих 2P-1 реализуется по другому. Если число не превышает значение 2P-1 из него вычитается модуль P,а результат является остатком.

Теперь на основе вышерассмотренного ФЧО можно построить матричную схему устройства приведения числа A по модулю P. На рисунке 3.3 приведена структурная схема матричной схемы приведения по модулю для чисел A= a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0 и P= P_3 P_2 P_1 P_0. Матричная схема работает следующим образом. По сигналу «Пуск» приводимое Число A принимается в регистр РгА, а разряды модуля P принимается в регистр РгР. Инверсный код модуля P и уровень +1 подается на входы всех формирователей остатков $ФЧО_0$-$ФЧО_3$ значения частичного остатка r= a_7 a_6 a_5 a_4 подается на вход $ФЧО_0$ и одновременно значение $ФЧО_0$ подается значения $2r_0$ из которого вычитается значение P путем сложения к rr_0 модуль P в дополнительном коде и вычисляется частичный остаток $r_1 = 8r_0 + P+1$.

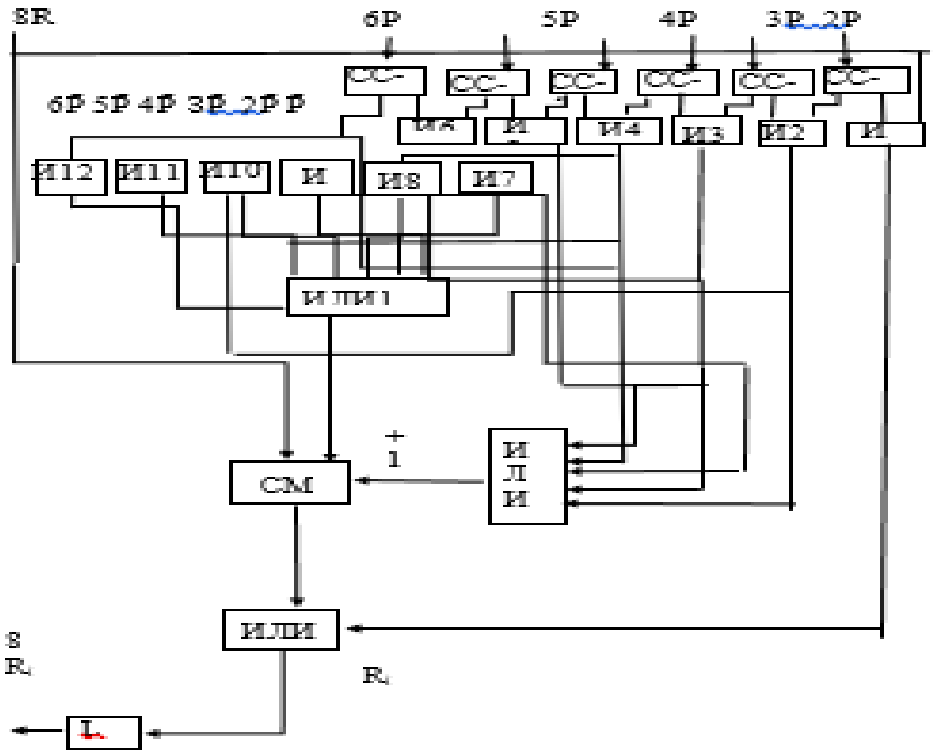

Рисунок 3.3 – Структура ФЧО

Результат через блок схем ИЛИ4 и ИЛИ1 передается в старшие разряды регистра РгА. При выполнения условий 8ri-1 ≥ 2p и 8ri-1 < 3p на выходе схемы И8 формируется единичный сигнал, который подается на вход схемы ИЛИ3 и схемы блока И10, на вторые информационные входы которого подаются разряды модуля. Разряды модуля через блок схем ИЛИ2 поступают на правые входы сумматора СМ2, а на вход младшего.

На рисунке 3.4 приведена функциональная схема конвейера для приведения по модулю 2n разрядного числа А на n–разрядный модуль Р. Конвейер состоит из N ступеней и каждый ступень состоит из единичных формирователей ФЧО и буферных регистров для частичных остатков Prri, буферных регистров для младших разрядов числа А, еще не вступивших в операцию и буферных регистров модуля Р – РгР, если каждое приведенное число А, имеет разный модуль.

В тактовом конвейерном устройстве приведения чисел по модулю, состоящем из N ступеней. Приводимые числа А, и его модуль Р, могут подаваться на входы с интервалом N раз меньшим, чем приведенная по модулю. В том числе также появляется и результаты на выходах конвейера.

Конвейер работает следующим образом. После подачи тактового импульса ТИ1 в регистры РгА и РгР принимаются первая пара А1 и Р1 чисел. При этом старшие a2n–1 ÷ an регистра РгА составляет частичный остаток – r0, а удвоенное значение 2r0 составляет разряды a2n–1 ÷ an–1 которые предаются вторым тактовым сигналом ТИ2 на входы ФЧО1 и вычисляется ri=2romodP, выполняя операцию ri=2r0+p+l, которая записывается в буферный регистр Prri.

При этом разряды an–1 ÷ a0 регистра РгА перепишется буферный

регистр первый ступени, а содержимое РгР перепишется в регистр РгР первый ступени. Тактовым импульсом ТИ2 в регистры РгА и Ргт также принимаются следующая пара чисел – А2 и Р2. После подачи третьего тактового импульса ТИЗ в регистры РгА и РгР принимаются пара чисел А3 и РЗ одновременно А2 и Р2 обрабатываются в ФЧО, и вычисляется и для этой пары и результат принимается в регистр Ргri. Необработанные разряды А2 записывается в буферный регистры первый ступень, а Р2 принимается в регистр РгР, передается на вход ФЧО2, где вычисляется значение r2=2ri modP и записывается в буферный регистр Ргг2 второй ступени и в регистрах второй ступени записываются не вступившие в операций разряды числа А1 и Р1.

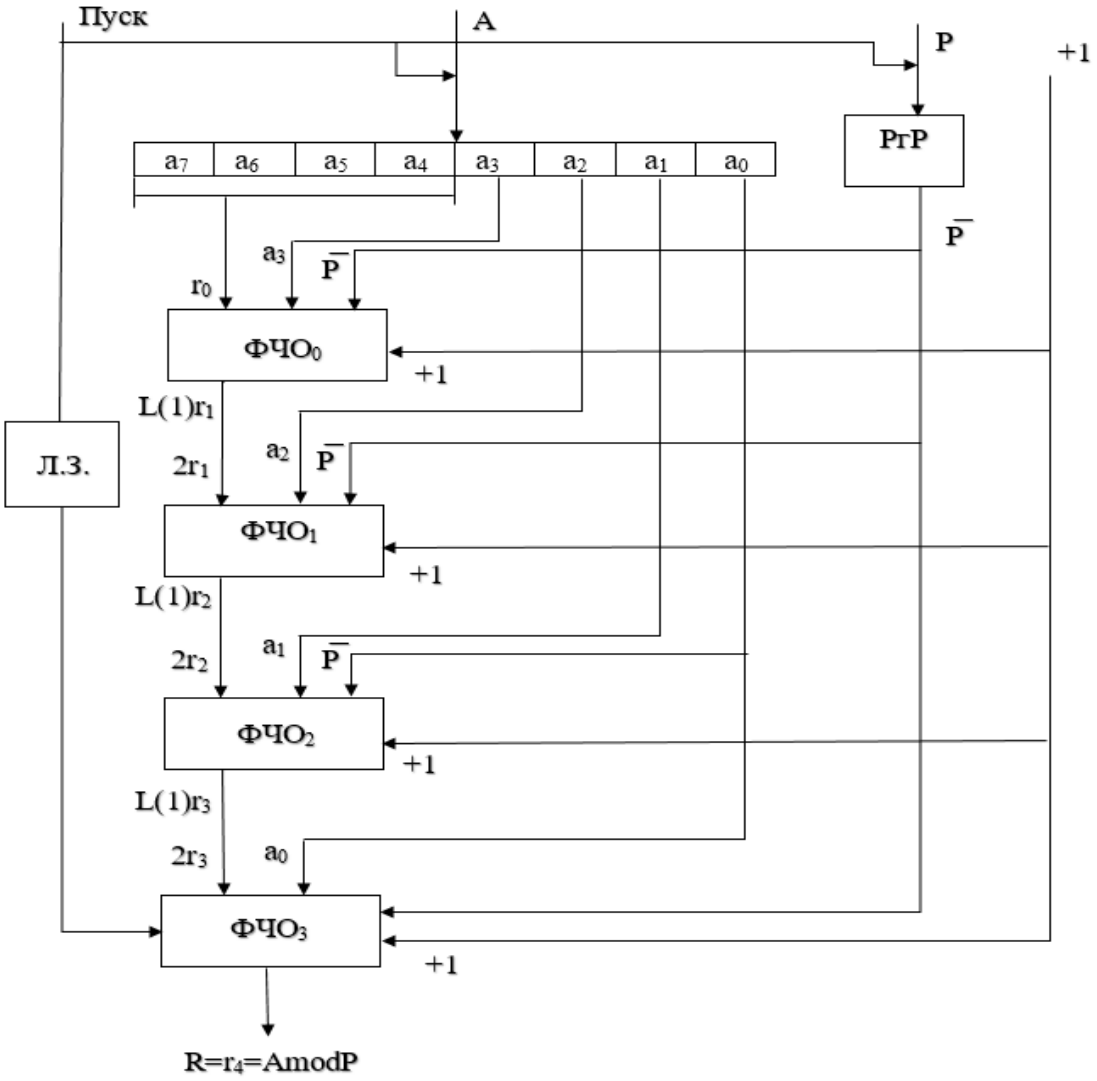

Рисунок 3.4 – Матричная схема приведения чисел по модулю

Время задержки Л.З. определяется суммированием прохождения кодов числа А через ФЧО0-ФЧОз. Время задержки Т на одном ФЧО-ТФЧО складывается тз времени задержки на сумматоре (τСМ) и мультиплексоре (τMS) т.е. τСМ tфчо = τСМ + τMS тогда ТЛ.З. n * Тфчо. По времени Тл.з.

определятся быстродействие матричной схемы приведения числа по модулю.

В матричной схеме приведения числа по модулю заложен очень важный потенциал повышения производительности – возможность конвейеризации. При конвейеризации весь процесс приведения по модулю.

Разбивается на последовательность законченных шагов. Каждый из этапов процедуры приведения по модулю выполняется на своей ступени конвейера, причем все ступени работает параллельно. Результаты полученные i-й ступени, передаются на дальнейшую обработку в (i+1)-ю ступень конвейера. Перенос информации со ступени на ступень происходит через буферную память, размещаемую между ними. Синхронность работы конвейера обеспечивается тактовыми импульсами, период которых τ определятся самой медленной ступенью конвейера τi и задержкой в элементе буферной памяти.

Конвейер состоит из трех ступеней. В каждый ступень состоит из: ФЧО; буферных регистров частичного остатка; регистров РгЗР и РгР; разрядов числа А, еще не вступивших в операцию. Конвейер управляется тактовыми импульсами ТИ. После заполнения конвейера при подаче каждого тактового импульса на выходе формируется результат пары Ai и Pi. В архитектуре множества вычислительных проектов где конъевиразация приносит ощутимый прирост вычислительных систем.По степени конвейеризации конвейеры могут синхронными и ассинхронными. Синхронными конвейерами обычно исползует в ВМ. Повысить производительность процессора можно за счет параллельного выполнения отдельных этапов рабочего цикла команд.

После подачи тактового импульса ТИN в регистре последней ступени Ргr2n-1. формируется результата R= rn–1.

После подачи ТИN+l, ТИN+2, ТИN+3 и тд. в регистре Ргrn-1 будет формирователем остатка от пары чисел А2 Р2, А3 Р3 и т.д. Результаты, вычисленные на i-й ступени на формирователе частичных остатков ФЧО i, передаются для дальнейшей обработки в (i+1) ступень конвейера. Перенос информации со ступени на ступень происходит через буферные регистры (БРг i), размещенные между ними. Выполнившая свою операцию i–я ступень помещает результат в i-й буферный регистр и может приступить к обработке следующей порции данных, в то время как очередная (i+1) ступень конвейера в качестве исходных использует данные, хранящиеся в i-м буферном регистре, расположенном на ее входе. Синхронность работы конвейера обеспечивается тактовыми импульсами (ТИ), период которых определяется самый медленной ступенью конвейера и задержкой в триггерах буферного регистра. В конвейерном устройстве приведения по модулю, имеющем К ступеней, входные данные могут подаваться на вход с частотой в К раз большей, чем в случае обычного делительного устройства. С этой же частотой будет появляться и результат на выходе устройства.

В тактовом конвейерном устройстве приведения чисел по модулю, состоящем из N ступеней. Приводимые числа А, и его модуль Р, могут подаваться на входы с интервалом N раз меньшим, чем приведенная по

модулю. В том числе также появляется и результаты на выходах конвейера.

Очень много схем включают в себя конвейеры в 7, 10 или даже 22 уровней. Новые ядра Pentium 5 с кодовыми именами Pesscott и Cedar Mill (и их Pentium T-производные) имеют 32 конвейер.

Многие процессора имеет конвейер длиной более чем в тысячу шагов. Недостатком в данном случае является необходимость сбрасывать весь конвейер в случае, если ход программы изменился (например, по условному оператору).Эту проблему пытаются решать предсказатели переходов. Предсказание переходов само по себе может только усугубить ситуацию, если предсказание производится плохо.

Конвейер состоит из трех ступеней. В каждый ступень состоит из:

- ФЧО;
- буферных регистров частичного остатка;
- регистров РгЗР и РгР; разрядов числа А, еще не вступивших в операцию.

Конвейер управляется тактовыми импульсами ТИ. После заполнения конвейера при подаче каждого тактового импульса на выходе формируется результат пары Ai и Pi.В архитектуре множества вычислительных проектов где конъевиразация приносит ощутимый прирост вычислительных систем.

Синхронность работы конвейера обеспечивается тактовыми импульсами (ТИ), период которых определяется самый медленной ступенью конвейера и задержкой в триггерах буферного регистра. В конвейерном устройстве приведения по модулю, имеющем К ступеней, входные данные могут подаваться на вход с частотой в К раз большей, чем в случае обычного делительного устройства.

Пусть А=63810=001001111102 Р=3510=0100112

Таблица 2 – Последовательность вычисления частичных остатков на ступенях конвейера

	ТИ1	ТИ2	ТИ3
РгА РгР	$A=001001111102_2$ $P=0100112_{\ 2}$		
$ФЧО_1$–ст.		$r_1=(8r_0+a_3)modP$ $8r_0+a_3$ 0.01001 $\overline{p}+1$ $\frac{1.10101}{1.11110}$ ЗН=1 поэтому r_1=0.01001	
$ФЧО_2$–ст.			$r=r=(2r_3+a_0)modP$ $8r3+a_0$ 0.01100

			$\overline{p}$ +1 1.10101 ЗН=0 и П=0 $r=r_2=0.01000$

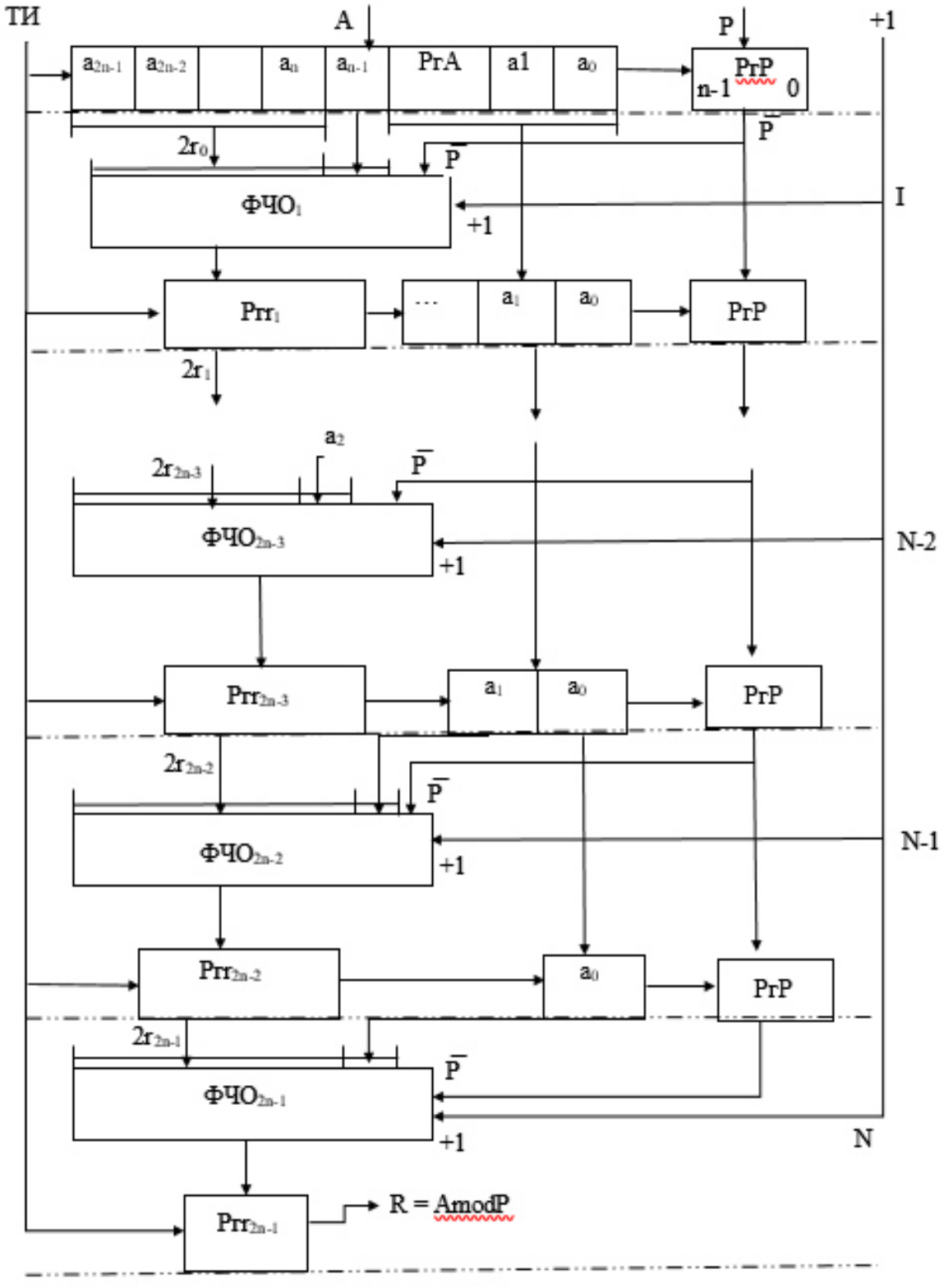

Рисунок 3.5 – Конвейеризованная матричная схема приведения числа по

модулю

4 Реализация алгоритма на языке программирования

Все процессы в машине на самом низком, аппаратном уровне приводятся в влияние только лишь только командами (инструкциями) машинного языка. Язык ассемблера – это символическое представление машинного языка. Ассемблер позволяет писать короткие и быстрые программы. Вобщем этот процесс довольно трудоёмкий. В последствие сего вполне вероятно на языке ассемблера пишутся в основном программки которые должны обеспечивать эффективную работу с аппаратной частью. Ещё на языке ассемблера пишутся опас по времени выполнения или же же расходованию памяти участки программки. Впоследствии чего они оформляются в виде подпрограмм и смешиваются с кодом на языке высокого смысла. Всякий процессор, в принципе имеет индивидуальный набор команд и соответствующий ему язык (или диалект) ассемблера. Связывание ассемблерного кода с другими языками

Главная множество современных компиляторов выделяют вероятность соединять в одной программе код, написанный на всевозможных языках программирования. Это позволяет быстро писать трудныепрограммки используя высокоуровневый язык, не теряя быстродействия в критических ко времени задачах, используя для их части написанные на языке ассемблера. Комбинирование можно достичь несколькими методами:

− на первом рубеже любой файл программки компилируется в объектный модуль;

− на втором объектные модули линкуются (связываются) в готовую программу.

− Регистры данных использоваться любым пользователем по своему желанию (за исключением некоторых случаев). В них можно хранить любые данные: числа, адреса и пр.[5].Заносить в стек или сохранять иным образом такие регистры, как a.bx,cx, dx, si, di и т. п., не обязательно, если вы, конечно, не оставляете в них информацию перед запуском программы, которая понадобится после завершения порожденного процесса.

Пришло время рассмотреть логические команды процессора, которые используются в файле-приложении. Логических команд всего несколько: and, or, xor.

Проще всего объяснить их действие на примерах. Главное — понять принцип. Мы еще не раз будем использовать логические операторы. Это одна из быстрых и простых вещей ассемблера, в отличие от языков высокого уровня.

− Оператор or (ИЛИ) служит для включения определенных битов;

− Оператор xor (исключающее ИЛИ) используется в основном для кодирования данных;

− Оператор and (И) служит для исключения битов.

Для использования нашего устройства будут также использованы обратный, прямой и дополнительный код.

Обратный и дополнительный код применяется в ЭВМ для операциями с отрицательными числами.Прямой код используется для записи положительных чисел.Обратный код способ представления чисел с фиксированный запятой позволяющего вычесть одно число из другого использую только операцию сложения.Дополнительный код это наиболее распространенный способ в современнах ЭВМ он также позволяют проводить операции с отрицатльными числами.

4.1 Установка программы

На рисунке 4.1 – 4.3 показаны установка программы:

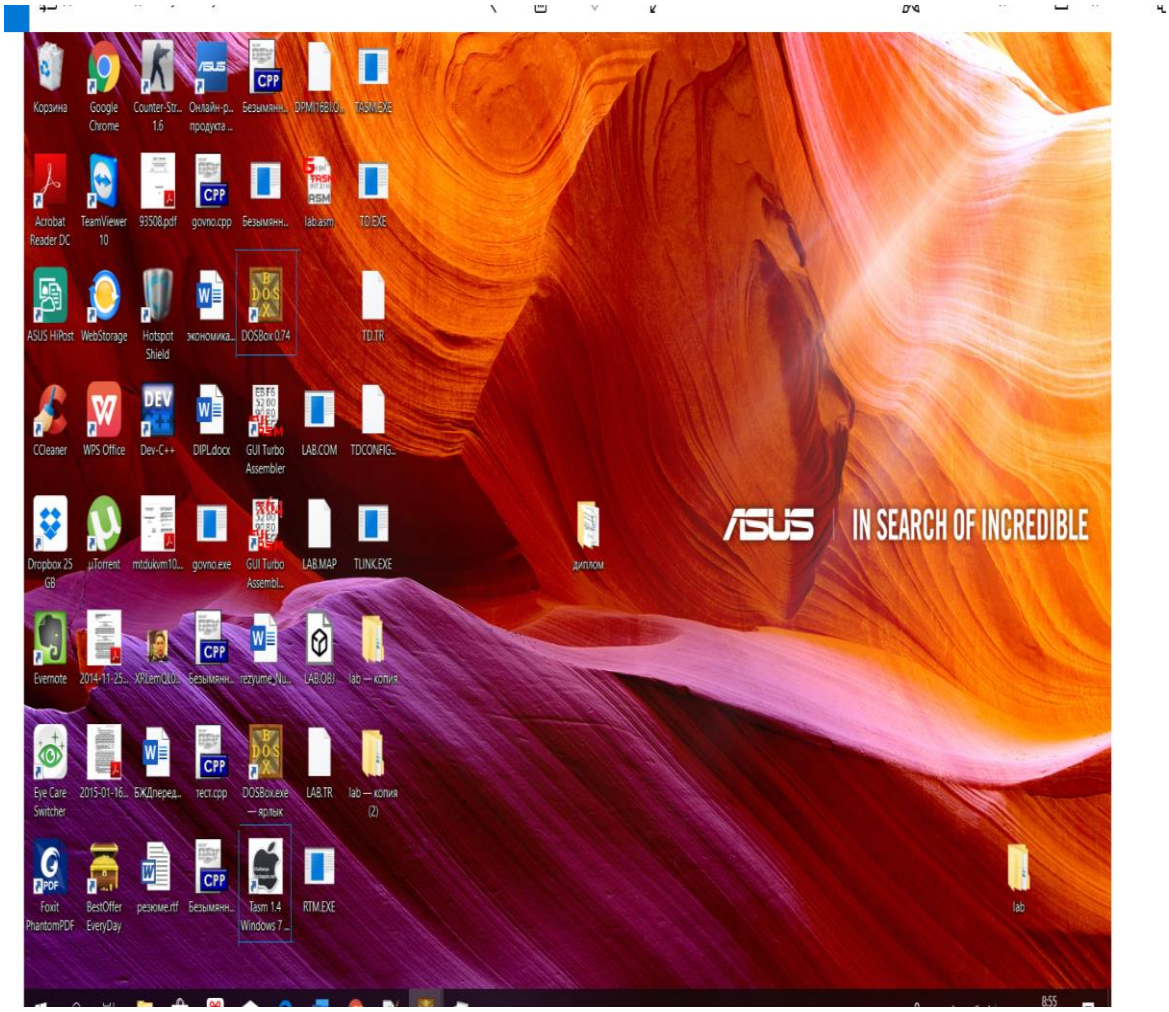

Рисунок 4.1 – Установка программы

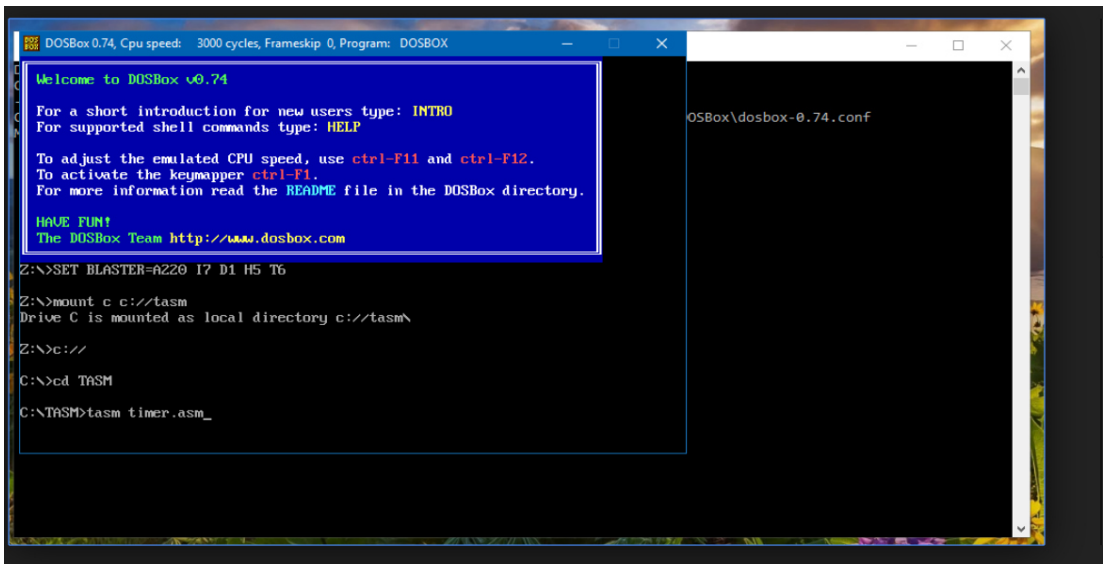

Рисунок 4.2 – Запуск программы

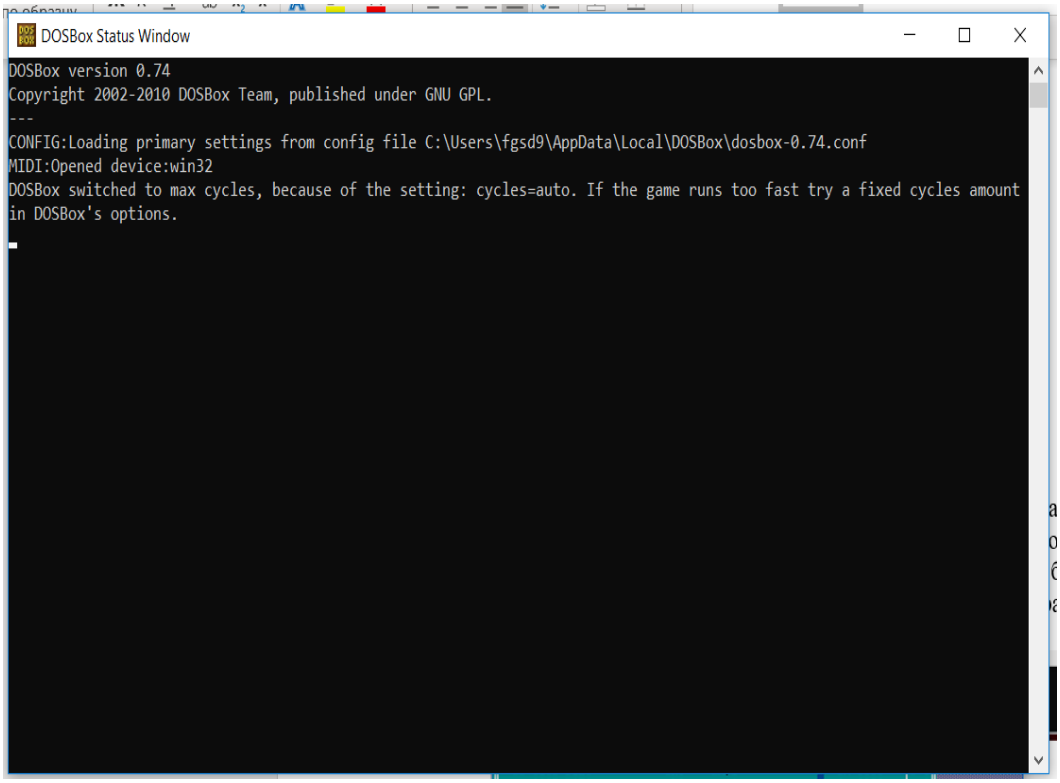

Рисунок 4.3 – Запуск ассемблера

Довольно нередко в программках применяется операция добавления или же вычитания единицы.

Прибавление единицы именуется инкрементом, а вычитание декрементом.Для данных операций естьособые команды микропроцессора: INC и DEC. Обратите забота, собственно что эти команды не изменяют смысл флага CF (Рисунок 4.4, 4.5).

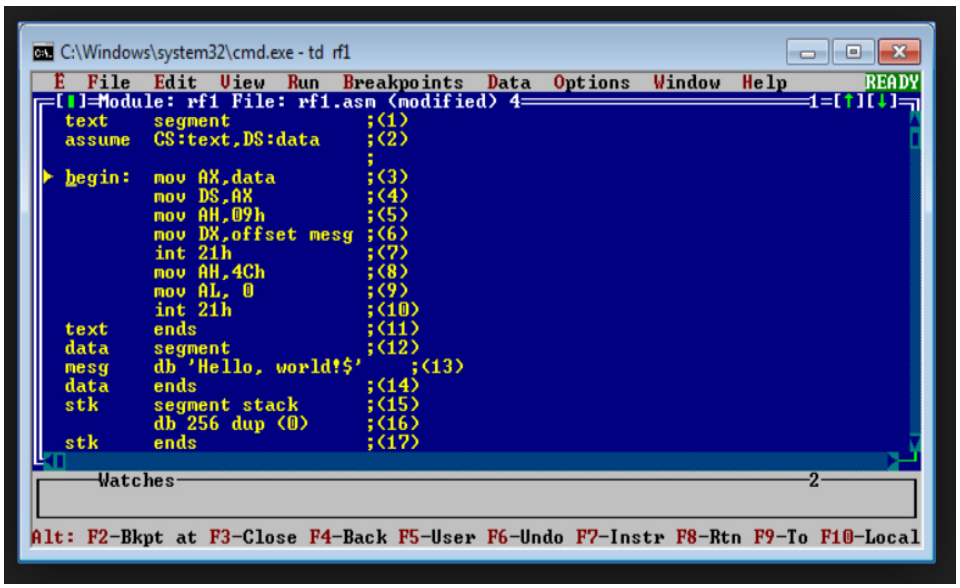

Рисунок 4.4 – Перевод в двоичных чисел

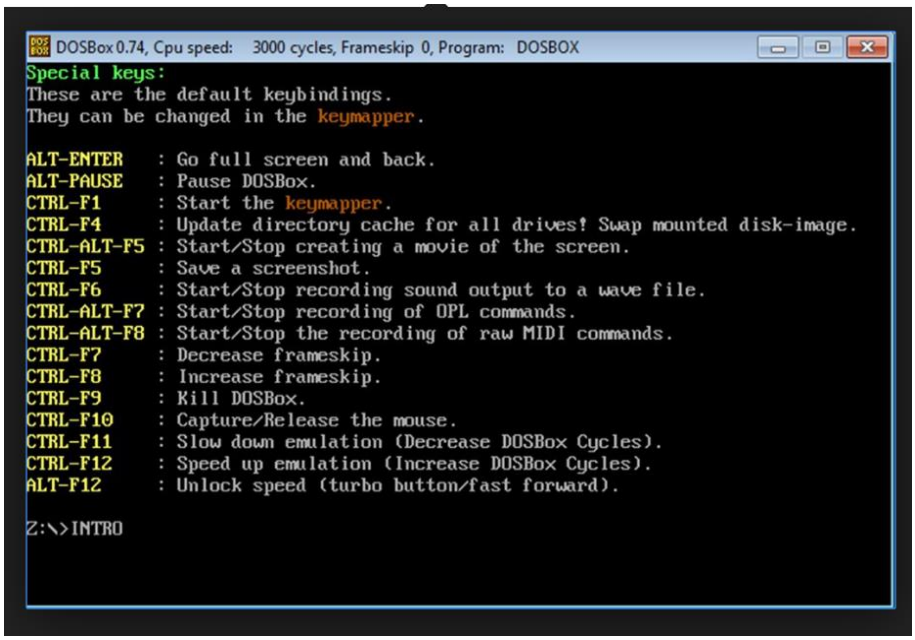

Рисунок 4.5 – Подсказки

Повторяющийся сдвиг выделяется от линейного тем, собственно что выдвигаемые с 1-го конца биты вдвигаются с иной стороны. В микропроцессора x32 есть 2 облика повторяющегося сдвига: незатейливый и сквозь флаг перенесения (CF). У всех команд, рассматриваемых в данной части дипломной работы, по 2 операнда, как у команд линейного сдвига. 1-ый операнд–сдвигаемое смысл и пространство для записи итога. 2 операнд-счётчик сдвигов, который имеет возможность располагается в регистре CL или же указываться именно.

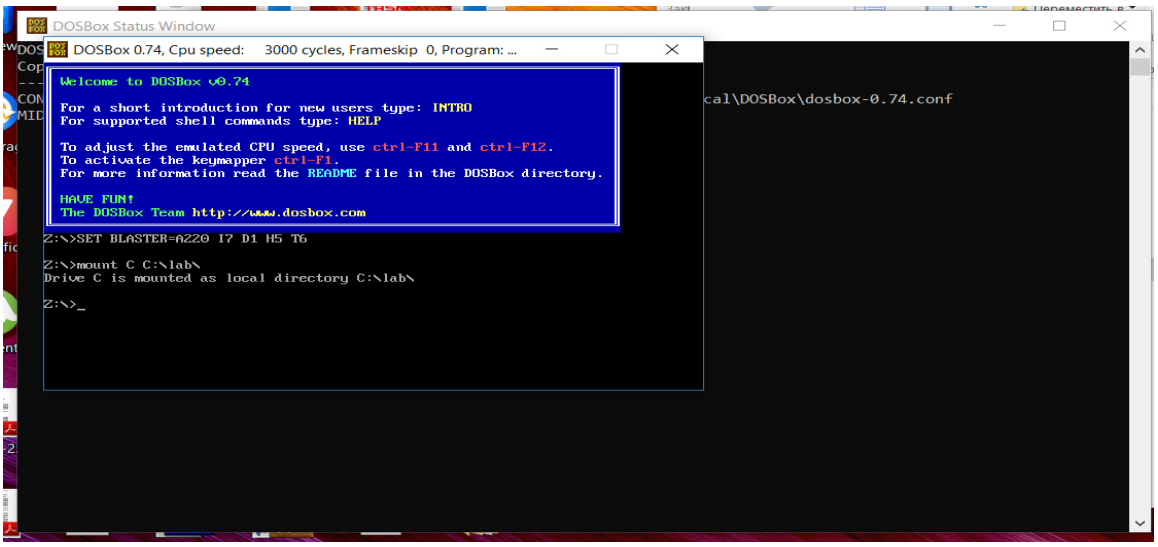

Рисунок 4.7 – Открытие кода

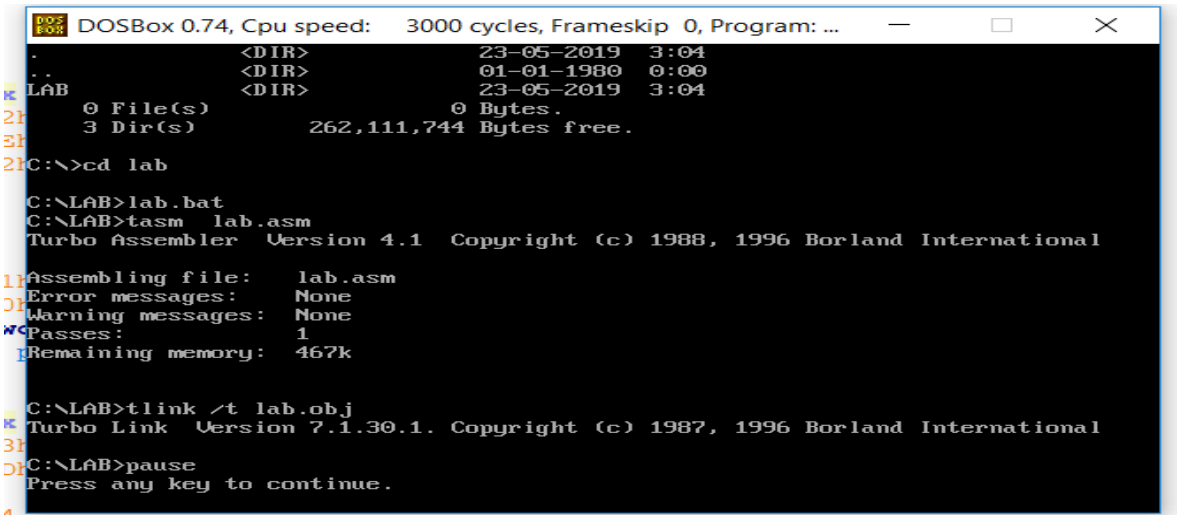

Рисунок 4.8 – Компиляция кода

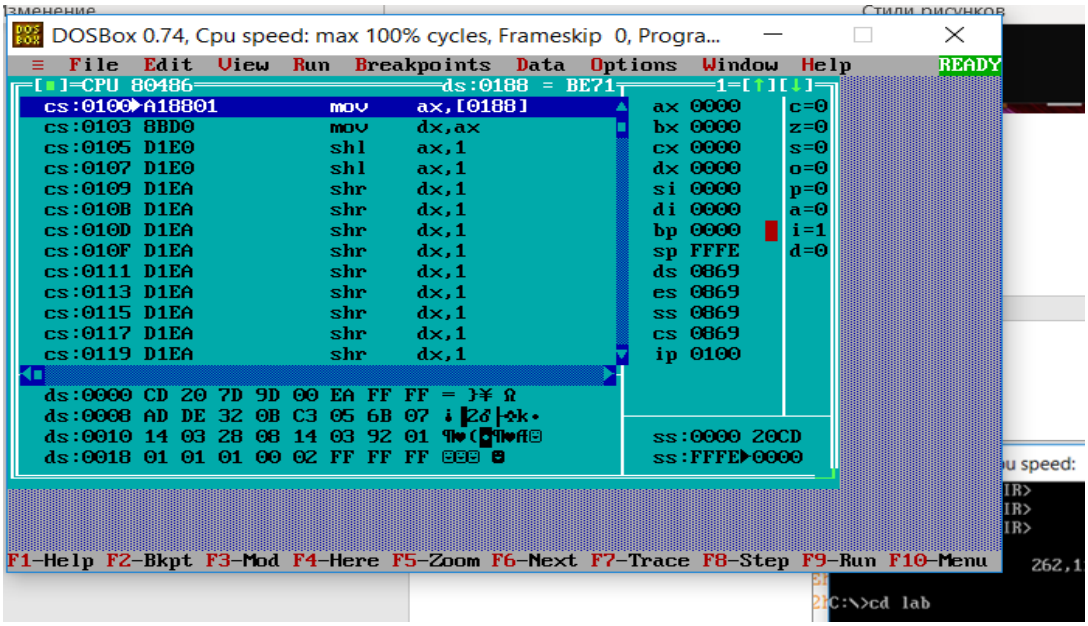

Рисунок 4.9 – Выводы результата

Для умножения чисел со знаком предназначена команда IMUL. Эта команда имеет три формы, различающиеся количеством операндов:

– с 1 операндом – форма, аналогичная команде MUL. В качестве операнда указывается множитель. Место другого множителя и результата определяется по таблице.

– с 2 операндами – указываются два множителя. Результат задается на место первого множителя. Большая часть результата в этом случае игнорируется. Кстати, эта форма команды не работает с операндами размером 1 байт.

– с 3 операндами – указывается положение результата, первого и второго множителя. Второй множитель должен быть непосредственным значением. Результат имеет такой же размер, как первый множитель, старшая часть результата игнорируется. Это форма тоже не работает с однобайтными множителями.

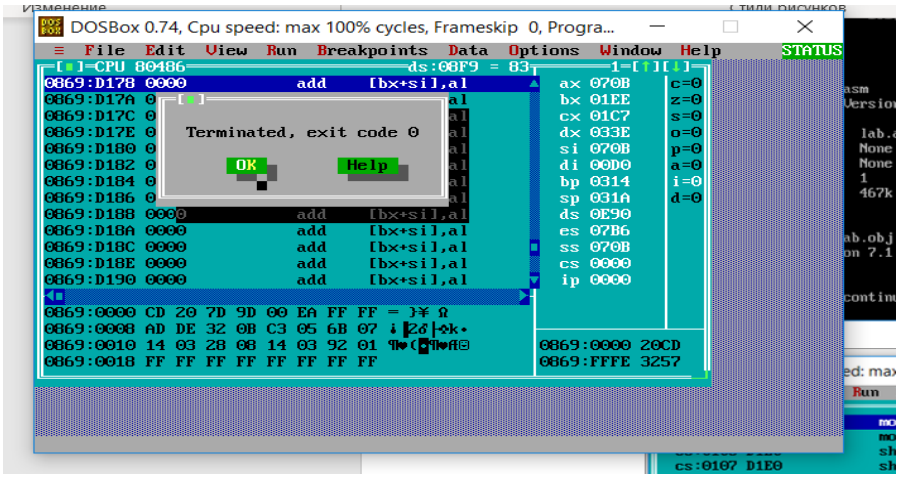

Рисунок 4.10 – Отладка кода

Деление цельных двоичных чисел–это всякий раз деление с остатком. Дележ количеств без символа исполняется с поддержкой команды DIV.

У данной команды раз операнд-делитель, который обязан пребывать в регистре или же в памяти. Месторасположение делимого, личного и остатка задаётся неявно и находится в зависимости от объема операнда:

– при выполнении команды DIV может возникнуть (о прерываниях я подробно расскажу потом, пока старайтесь избегать таких случаев): если делитель равен нулю;

– если частное не помещается в отведённую под него разрядную сетку (например, если при делении слова на байт частное больше 255).

Для деления чисел со знаком предназначена команда IDIV. Единственным операндом является делитель. Местоположение делимого и частного определяется также, как для команды DIV. Эта команда тоже генерирует прерывание при делении на ноль или слишком большом частном.

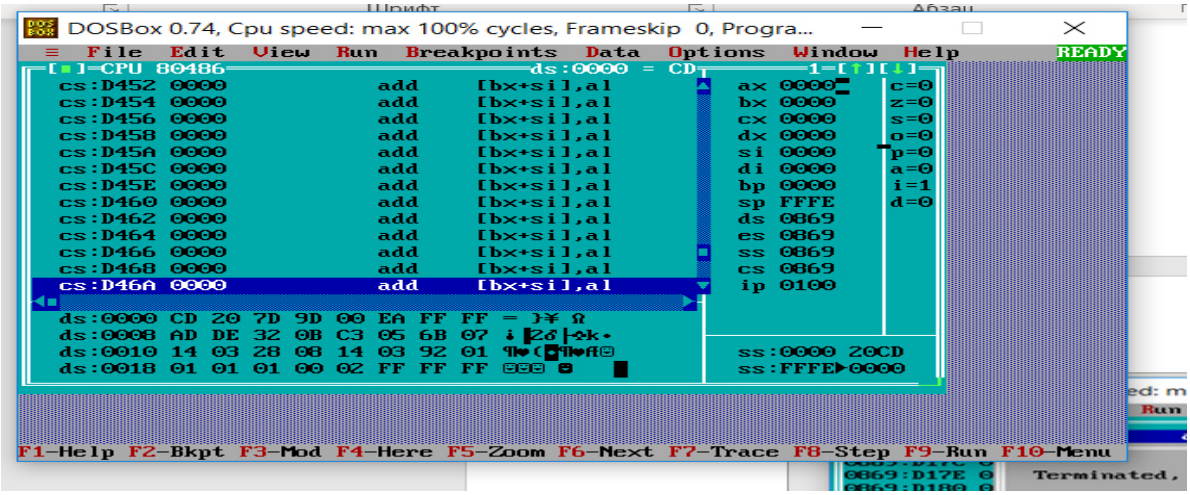

Рисунок 4.11 – Выводы

4.2 Блок схема программы

На рисунке 4.12 приведен алгоритм деления с со сдвигом на 3 разряда:

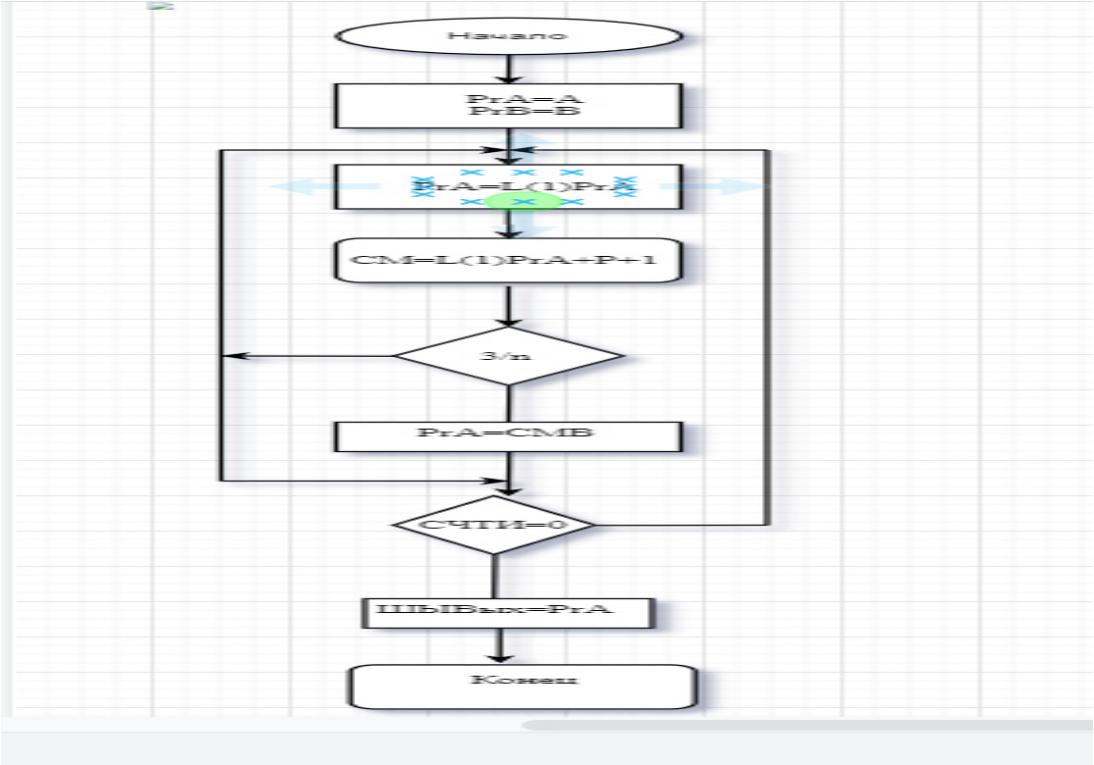

Рисунок 4.12– Алгоритм деления с со сдвигом на 3 разряда

5 Глава безопасность жизнедеятельности

5.1 Анализ условий труда

В дипломной работе я проектирую систему шифрования. В ее эксплуатации необходим особо секретный объект , в котором будет работать 4 человека, сотрудник охраны и люди обладающие навыками в области информационной безопасностью . В помещении есть несколько рабочих мест с установленным на нём компьютером и мониторами и аппаратным устройством для шифрованием .Данном помещение присуствует современная шумоподавление и вентиляция .В разделе БЖД задаюсь целью рассчитать естественное и искусственное освещение.

Таблица 5.1– Исходные данные

Тип помещения	Параметры помещения				Разряд зрит. работ	$\rho_{пот}$	$\rho_{стен}$	$\rho_{пол}$	$h_{нок}$, м	Световой пояс	Нзд	Расст. до рядом стоящего здания, Р
	, м	,м	,м	, м								
Помещения для шифрования	1	5			III,б	0	0	0		Алматы	8	10

Расчетная часть

Расчет естественного освещения

Изначально в помещение в котором мы рассматривали было 15 ламп и окно 60м2

Расчет площади световых проемов

Площадь боковых проемов при боковом освещении определяется из следующей формулы:

$$100 \cdot \frac{S_0}{S_n} = \frac{e_N \cdot К_з \cdot \eta_0}{\tau_0 \cdot r_1} \cdot \quad (5.1)$$

где S_0– площадь световых проемов при боковом освещении, м2;
Sn– площадь пола помещения, м2;
eN – нормируемое значение КЕО;
КЗ –коэффициент запаса;
η_0– световая характеристика окон;

τ_0– общий коэффициент светопропускания;

r_1– коэффициент, учитывающий повышение КЕО при боковом освещении, благодаря свету, отраженному от поверхности помещения и подстилающего слоя, примыкающего к заданию;

$К_{зд}$ – коэффициент, учитывающий затемнение окон противостоящими зданиями.

Определим площадь пола помещения:

$$S_n = L \cdot B \tag{5.2}$$

Нормируемое значение КЕО, eN, для заданий, располагаемых в различных районах определять по формуле:

$$e_N = e_H \cdot m_N$$

где mN – коэффициент светового климата, определяемый из таблицы

Учитывая заданный световой пояс (г.Алматы) адм. район 9, приняв ориентацию световых проемов З , В определим:

m_N= 0.8

e_H – значение КЕО.

По таблице1.3 , учитывая III б разряд зрительных работ, найдем:

e_H=1.2

Следовательно:

$$e_N = 1.2 \cdot 0.8 = 0{,}96 \tag{5.3}$$

Учитывая тип помещения, найдем коэффициент запаса с помощью таблицы 10. каб.учебн. помещение, лаб. раб.

Кз=1.2 при ЕО вертикально.

Для определения световой характеристики, η_0, необходимо рассчитать отношение длины помещения к его глубине $\frac{L}{l}$, отношение ширины помещения к расчетной высоте

$$\frac{l}{h_{расч}}.$$

$$l = B - 1 = 15 - 1 = 14 \text{ м} \tag{5.4}$$

$$\frac{L}{l} = \frac{21}{14} = 1.5$$

Найдем h расчет:

$$h_{расч} = h_{0к} + h_{н.ок.} - h_{р.п.}$$

$$h_{расч} = 4 + 1 - 0.8 = 4.2 \text{ м}$$

$$\frac{l}{h_{рас}} = \frac{14}{4.2} = 3.3 \approx 3$$

$$\frac{l}{B} = \frac{14}{15} = 0.93 \approx 1$$

Учитывая найденные отношения примем световую характеристику, $\eta_0 = 15$, по таблице 2 .

Общий коэффициент светопропускания, τ_0, рассчитывают по формуле:

$$\tau_0 = \tau_1 \cdot \tau_2 \cdot \tau_3 \cdot \tau_4, \quad (5.5)$$

где τ_1 – коэффициент светопропускания материала, принимаемый по таблице 4. Так как в качестве светопропускающего материала используется стекло листовое двойное, то:

$\tau_1 = 0.8$

τ_2 – коэффициент, учитывающий потери света в переплетах светопроема. Определяется с помощью таблицы 5 с учетом использования стальных двойных глухих переплетов:

$\tau_2 = 0.8$

τ_3 – коэффициент, учитывающий потери света несущих конструкциях, при ,боковом освещении:

$\tau_3 = 1$

τ_4 – коэффициент, учитывающий потери света в солнцезащитных устройствах, принимается по таблице 5.2. Выбираем убирающиеся регулируемые жалюзи и шторы (межстекольные внутренние, наружные)

$\tau_4 = 1$

Следовательно:

$$\tau_0 = \tau_1 \cdot \tau_2 \cdot \tau_3 \cdot \tau_4 = 0.8 \cdot 0.8 \cdot 1 \cdot 1 = 0.64$$

$$\rho_{ср} = \frac{\rho_{пот}\rho_{стен}\rho_{пол}}{3}\% = \frac{50 + 50 + 10}{3} = 0{,}36 \approx 0{,}3$$

r1=1,7

Учитывая Нзд=18 и Р=10 м (расстояние до рядом стоящего здания) из таблицы 9 найдем коэффициент, учитывающий затемнение окон противостоящими зданиями, Кзд:

Таблица 5.2– Исходные данные

Тип помещения	Параметры помещения				Разряд зрит. работ	$\rho_{пот}$	$\rho_{стен}$	$\rho_{пол}$		Ф
	,м	,м	,м	, м						
Помещение для шифрования	1	5			III,б	0	0	0	6	5000 лм

$$\frac{P}{H_{зд}} = \frac{10}{18} = 0.55 => К_{зд} = 1.7$$

Зная значение всех параметров, рассчитываем площадь боковых проемов при естественном освещении по следующей формуле:

$$S_0 = \frac{S_n \cdot e_N \cdot К_з \cdot \eta_0}{100 \cdot \tau_0 \cdot r_1} \cdot К_{зд}$$

$$S_0 = \frac{315 \cdot 0{,}96 \cdot 1.2 \cdot 15 \cdot 1{,}7}{100 \cdot 0.64 \cdot 1{,}7} = 85{,}05 \text{ м}^2$$

Таким образом данных рассчетов ествественное освещение не удовлетворяет рассчитаному нормативному значения.

5.2 Расчет искусственного освещения

Для расчета искусственного освещения используют один из трех методов: по коэффициенту использования светового потока, точечный и метод удельной мощности. При рас чете общего равномерного освещения основным является метод использования светового потока, создаваемого источником света, и с учетом отражения от стен, потолка, пола.

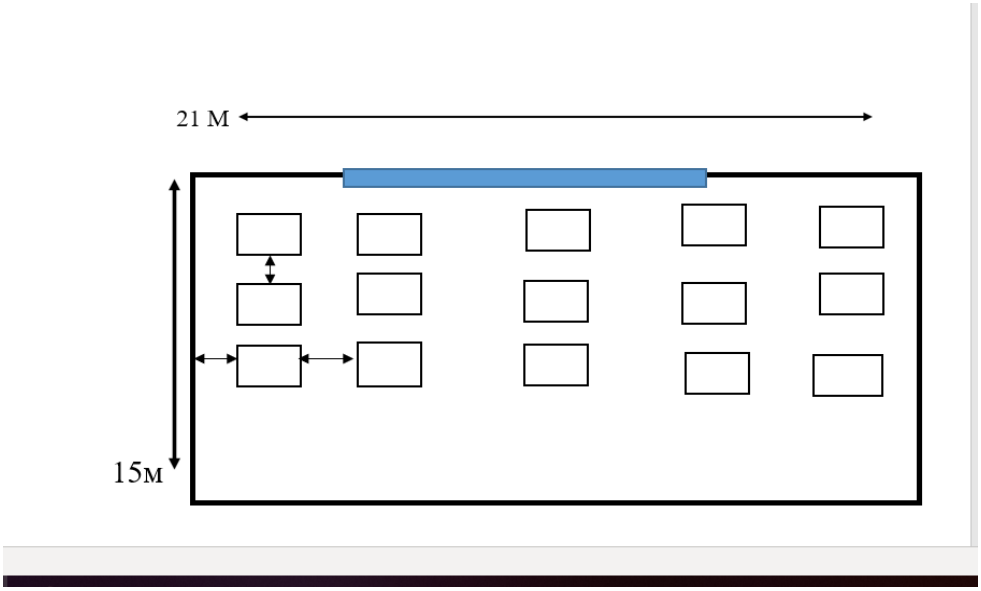

Рисунок 5.1 – Помещения

Расчет освещения начинают с выбора типа светильника, который принимается в зависимости от условий среды и класса помещений по взрыво-пожароопасности.

$$E_\tau = \frac{Nn\,\phi\cdot\mu}{K\cdot S\,V} = \frac{15*2*5000*0{,}55}{1.5*315*1.1} = 158;$$

Сначала нужно рассчитать заданное номинальное значение оно должно быть больше 300.

Получилась у нас 158<300 что не удовлетворяет условному значению

Разряд зрительной работы III, б, поэтому нормируемая освещенность по таблице Ен= 300 лк (при системе общего освещения).

Определение расчетной высоты подвеса:

$$h_{\text{расч}} = H_{\text{помещения}} - H_{\text{свеса}} - H_{\text{р.п.}}, \qquad (5.6)$$

где Нсвеса = 0,5 – высота свеса ламп, м;

Нр.п. = 0,8 – расстояние рабочей поверхности над полом, м;

$H_{\text{помещения}} = 6$ – высота помещения, м.

$$h\text{расч}=6–0{,}5–0{,}8=4{,}7 \text{ м};$$

В практике расчетов значения коэффициентов η находятся из таблиц, связывающих геометрические параметры помещения (индекс помещения) с их оптическими характеристиками.

Индекс помещения определяется по формуле :

$$i = \frac{A*B}{h_{\text{расч}}*(A+B)} = \frac{21*15}{4{,}7(21+15)} = \frac{315}{169{,}2} = 1{,}86; \qquad (5.7)$$

где А– длина помещения , м;
В– ширина помещения , м;
$h_{\text{расч}}$ – расчетная высота, м.

По таблице 15 для светильника типа TLPL228.2x36 находим η = 0,55. Таким образом, количество светильников равно:

$$N = \frac{\text{Е}_\tau \cdot \text{К}_\text{З} \cdot \text{S} \cdot \text{V}}{n\ \text{ф} \cdot \mu} = \frac{300 \cdot 1.5 \cdot 315 \cdot 1.1}{2 * 5000 * 0{,}55} = 28$$

Ен = 300 лк – заданное номинальное освещение.
S = 315м2 – площадь помещения.
Z = 1,1– коэффициент неравномерности освещения.
n– количество ламп в светильнике.
ф = 5000 лм

$$N = \frac{200 * 1{,}5 * 242 * 1{,}1}{0.44 * 2 * 3120}$$

$$N \approx 28\text{шт}$$

5.3 Расчет освещенности точечным методом

Определим расчетную высоту подвеса.

$$\text{hр}=\text{Нп}-\text{hсвеса}-\text{hр.пов.}$$

Принимаем hсвеса = 0,5 м hр.пов.= 0,8 м

$$\text{Нр} = 6-0{,}5-0{,}8 = 4{,}7\ \text{м}$$

Найдем расстояние между светильниками, учитывая λ=0.6÷1.5.

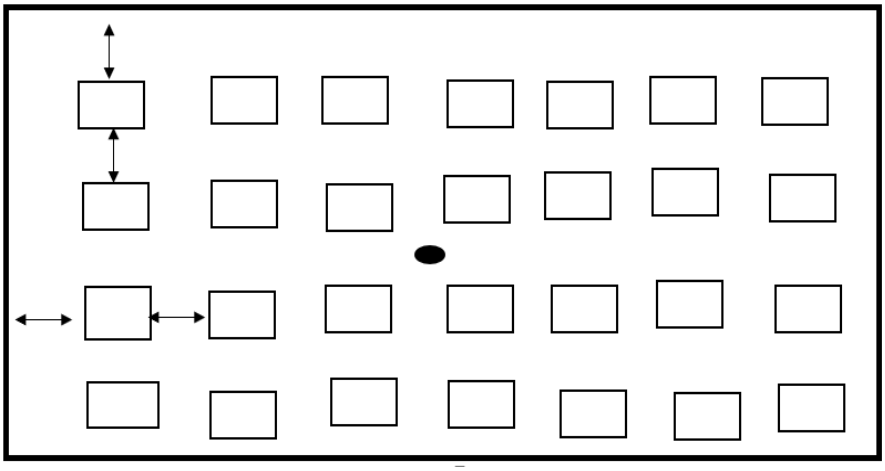

Рисунок 5.2 – Рассчет точечным методом

$$LA=\lambda$$

$$d7,9,12,14=\sqrt{4,5^2+2^2}=4,9\text{м}$$

$$d3,18=6\text{ м};$$

$$d8,13=2\text{ м};$$

$$hp=1,216\cdot 4,7=4,5\text{ м}$$

$$LB=LA-(0.3\div 0.5)=4,5-0,5=4\text{ м}$$

$$la=lb=(La/3)=4,5/3=1,5$$

$$lb=(La/3)=4,5/3=1,5$$

Для расчета намечаем контрольную точку А. Необходимо найти d3,18; d1,5,16,20; d8,13; d2,4,17,19; d6,10,11,15; d7,9,12,14; – проекции расстояния на потолок между точкой А и соответствующим светильником:

$$d1,5,16,20=\sqrt{6^2+9^2}=10,82\text{ м};$$

$$d2,4,17,19=\sqrt{4,5^2+6^2}=7,5\text{ м};$$

$$d6,10,11,15=\sqrt{2+9^2}=9,22\text{ м}$$

$$d7,9,12,14=\sqrt{4,5^2+2^2}=4,9\text{м}$$

$$d3,18=6\text{ м};$$

$$d8,13=2\text{ м};$$

Далее определяем угол между высотой потолка и соответствующим отрезком d:

$$\text{tg}\alpha 1=\frac{d_{1,5,16,20}}{h\text{расч}}=\frac{10,82}{4,7}=2,9\rightarrow \alpha 1=71°;$$

$$\cos^3\alpha_1=0,0345$$

$$\text{tg}\alpha 2=\frac{d_{2,4,17,19}}{h_{\text{расч}}}=\frac{7,5}{4,7}=2\rightarrow \alpha 2=64\ °;$$

$$\cos^3\alpha_2=0,084$$

$$\text{tg}\alpha 3=\frac{d_{6,10,11,15}}{h\text{расч}}=\frac{9,22}{4,7}=2,5\rightarrow \alpha 1=68\ °;$$

$$\cos^3\alpha_3=0,053$$

$$\text{tg}\alpha 4=\frac{d_{7,9,12,14}}{h\text{расч}}=\frac{4,9}{4,7}=1,3\rightarrow \alpha 1=53°;$$

$$\cos^3\alpha_4=0,218$$

$$\text{tg}\alpha 5 = \frac{d_{3,18}}{h\text{расч}} = \frac{6}{4,7} = 1,62 \rightarrow \alpha 1 = 58\ °;$$

$$\cos^3 \alpha_5 = 0,149$$

$$\text{tg}\alpha 6 = \frac{d_{8,13}}{h\text{расч}} = \frac{2}{4,7} = 0,54 \rightarrow \alpha 1 = 29\ °;$$

$$\cos^3 \alpha_6 = 0,669$$

Выбираем тип светильника ПВЛМ (с 2 лампами ЛБР)

Таблица – 3 Светотехнические характеристики светильника

Тип светильника	Освещеность I_α, кд при угле α									
	0	15	25	35	45	55	65	75	85	90
ПВЛМ	175	165	148	130	110	70	60	30	20	–

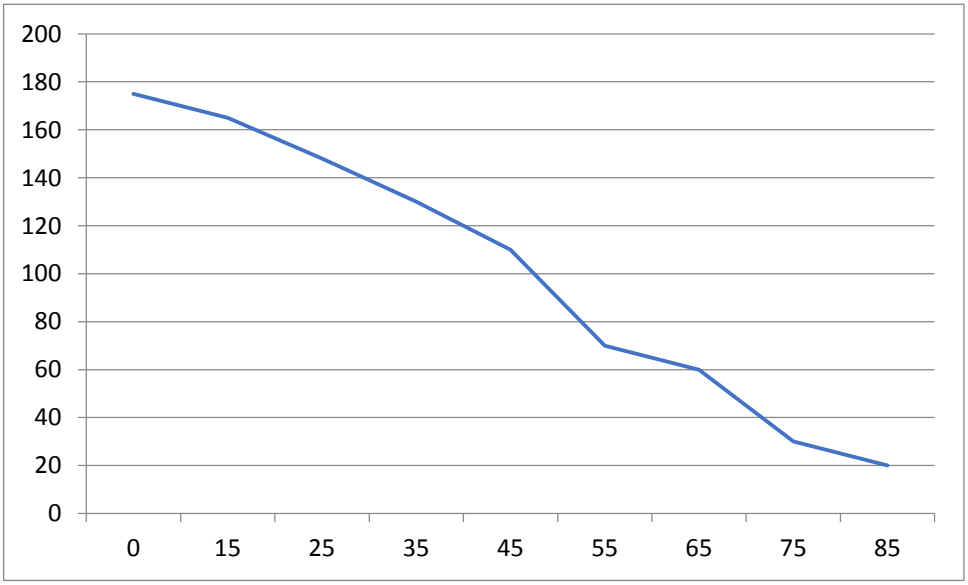

Рисунок – 5.3 Зависимость α=f(I_α)

По этому углу находим силу света от каждого источника по рисунку 1:

Iα1(1,5,16,20)=45 кд
Iα2(2,4,17,19)=61 кд

Iα3(6,10,11,15)=55 кд
Iα4(7,9,12,14)=77 кд
Iα5(3,18)=70 кд
Iα6(8,13)=140 кд

Освещенность помещения относительно контрольной точки от каждого источника:

$$e_{АГ} = \frac{n \cdot I_{\alpha} \cos^3 \alpha}{h_p}$$

$$e_{АГ1} = \frac{4 \cdot 45 \cdot 0{,}0345}{4{,}7^2} = 0{,}45\text{лк};$$

$$e_{АГ2} = \frac{4 \cdot 61 \cdot 0{,}084}{4{,}7^2} = 1{,}5\text{лк};$$

$$e_{АГ3} = \frac{4 \cdot 55 \cdot 0{,}053}{4{,}7^2} = 0{,}85\text{лк};$$

$$e_{АГ4} = \frac{4 \cdot 77 \cdot 0{,}218}{4{,}7^2} = 5\text{лл};$$

$$e_{АГ5} = \frac{2 \cdot 70 \cdot 0{,}149}{4{,}7^2} = 1{,}6\text{лк};$$

$$e_{АГ6} = \frac{2 \cdot 140 \cdot 0.669}{4{,}7^2} = 13{,}9\text{лк};$$

$$\sum_{i=1}^{15} e_{АГi} = e_{АГ1} + e_{АГ2} + e_{АГ3} + e_{АГ4} + e_{АГ5} + e_{АГ6} = 0{,}45+1{,}5+0{,}85+5+1{,}6+13{,}9=23{,}3\text{лк}$$

Суммарная освещенность:

$$E = \frac{\mu \cdot Ф_л \cdot n}{1000 \cdot K_з} \cdot \sum_{i=1}^{15} e_{АГi}$$

где μ – коэффициент, учитывающий действие «удаленных» светильников (1,1 ÷ 1,25). Световой поток выбранной лампы ЛБР(ЛХБР80) 4160 лм.

$$E_{АГ} = \frac{1.2 \cdot 4160 \cdot 40}{1000 \cdot 1.5} \cdot 23{,}3 = 464{,}521\text{лк}$$

Emin=300 лк, берем из таблицы 3.12. ЕАГ≥Emin(т.к. освещенность незначительно больше нормированного освещения нужно увеличить количетво светильников до 28 шт).

6 Технико-экономическое обоснование

Цель моего дипломного проекта заключается в разработке инновационного решения , позволяющего в максимально короткие сроки зашифровать и дешифровать ,при этом минимально сократить нагрузку на аппаратную систему. На данный момент существует множество различных схемотехнических решений обработки информации.В разработке инновационного решения будут специалисты, которая включает в себя: руководитель проекта , программист-разработчик аппаратный криптограф. Руководитель должен проверять рабочих графиков и их соблюдение. Обязанность программиста-разработчика входит разработка технического обоснования, разработка программного обеспечения, его тестирование и сопровождение.

Обязанность аппаратного криптографа входит шифровании и дешифрования информации. Технико-экономическое обоснование содержит следующие пункты:

- определение трудоемкость разработки программного обеспечения;
- расчет затрат на разработку ПО;
- определение ценности готового продукта.

6.1 Определение сложности разработки ПО

Чтобы точно определить трудоемкость разработки программного обеспечения, необходимо разделить задачу на этапы .. Модель распределения трудность разработки ПО и стадии разработки представлены в таблице 6.1.

Таблица 6.1 – Этапы разработки ПО

Этапы разработки ПО	Вид работы	Трудоемкость, чел. Час.
Этап 1	Постановка задач	15
Этап 2	Поиск и изучение программных продуктов	15
Этап 3	Поиск и изучение технической литературы	15
Этап 4	Поиск и изучение технической литературы	40
Этап 5	Реализация проекта	50
Этап 6	Отладка	40
Этап 7	Составление отчета о результатов работы	20
Этап 8	Тестирование продукта	20
Итого:		215

Продолжительность рабочего дня равна 8 часам. В результате для

разработки и реализации программного обеспечения необходимо 27,75=28 рабочих дней.

6.2 Расчет затрат на разработку ПО

Определение затрат необходимых для разработки программного обеспечения производится на основе имеющейся информации, которая включает следующие элементы:

– материальные затраты;
– затраты на оплату труда;
– социальный налог;
– амортизация основных фондов;
– прочие затраты.

Материальные затраты делятся на основные и вспомогательные затраты на материалы, энергию и другие затраты необходимые для разработки ПО. Расчет материальных затрат происходит по форме, предоставленной в таблице 6.2.

Таблица 6.2 – Затраты на материальные ресурсы

Наименование материала	Марка	Ед. измерения	Количество	Цена за ед. в тенге	Сумма в тенге
Бумага для офиса	International Paper	Упаковка	3	1 000	3 000
Тетрадь (96 листов)	tetrapak	Штук	2	190	380
Суммато р	Dori	Штук	2	400	2000
Ручки	Hauser	Штук	2	90	180
Компьютерная мышь	MICK	Штук	1	3 000	3 000
Итого:					8560

Для разработки программного обеспечения будет использоваться ноутбук ASUS X541U мощности ноутбука хватит для поставленной работы .

Общую сумму, необходимую на материальные средства ($З_м$) можно рассчитать по следующей формуле:

$$З_м = \sum P_i * Ц_i, \quad (6.1)$$

где P_i – расход i-го вида материального ресурса, натуральные единицы;

$Ц_i$ – цена за единицу i-го вида материального ресурса, тг;

i – вид материального ресурса;

n – количество видов материальных ресурсов.

Расчет затрат на необходимое оборудование и программное обеспечение производится по форме, приведенной в таблице 6.3.

Таблица 6.3 – Расчет затрат на оборудование и ПО, необходимое для проекта

Наименование материала	Марка	Ед. измерения	Количество	Цена за ед. в тенге	Сумма в тенге
Ноутбук	ASUS ZenBOOK	Штук	1	390 000	390 000
Принтер	HP 1183400	Штук	1	30000	30000
Устройства кодирования информации	MLDEZ	Штук	3	2 350	6 700
Модем	Toshiba 77RQ	Штук	1	14 000	14 000
ОС	Windows 7	Штук	1	–	–
Шифратор	XSLFRE	Штук	1	3 338	3 338
Итого:					444 038

$$Зм = 8560 + 444\,038 = 452\,598 \text{ (тг)}$$

Для разработки программного обеспечения необходимы материалы на сумму 452 598 тенге.

6.3 Расчет затрат на электроэнергию

Так как при разработке программного обеспечения не обойтись без потребления электроэнергии, имеет смысл произвести расчет затрат на электроэнергию.

Согласно таблице 6.1 для разработке программного обеспечения необходимо порядка 215 часов, теперь необходимо рассчитать стоимость электроэнергии, которая будет потрачена в течении 215 часов. Для принтера расчет будет проводиться для периода в 24 часа, так как нет необходимости постоянно использовать принтер.

$$Э = З_{эл.эн.обор.} + З_{доп.нужды.}, \quad (6.2)$$

где $З_{эл.эн.обор.}$ – затраты на электроэнергию оборудования;

$З_{доп.нужды.}$ – затраты электроэнергии на дополнительные нужды.

Расчет электроэнергии, которая необходима для оборудования определяется по следующей формуле:

$$З_{эл.эн.обор.} = \sum W * К_{исц} * S * T, \quad (4.3)$$

где W – потребляемая мощность, Вт;

$К_{исц}$ – коэффициент использования ($К_{исц}$ = 0,7..0,9);

Т – время работы;

S – тариф (1кВт/ч = 23,81 тг).

Итоги по расчетам стоимости затрачиваемой электроэнергии представлены в таблице 6.4.

Таблица 6.4 – Затраты на электроэнергию

Наименование приборов	Паспортная мощность, кВт	Коэффициент мощности	Время работы оборудования, ч	Цена ЭЭ тг/кВтч	Сумма, тг.
Ноутбук	0,6	0,7	215	23,81	2 254,9
Устройства аппаратного значения	0,08	0,7	215	23,81	40,64
Принтер	0,6	0,9	24	23,81	386,4
Освещение	0,3	0,7	215	23,81	1 127
Итого:					3808

$$З_{эл.эн.обор.} = 3808 \text{ (тенге)},$$

Дополнительные расходы подсчитываются на основе повышенного показателя в объеме 5% от расходов на электроэнергию:

$$З_{доп.нужды} = 5\% * З_{эл.эн.обор.}, \quad (6.4)$$

Определим затраты на дополнительные потребности согласно формуле (4.4):

$$З_{доп.нужды} = 0.05 * 3808 = 190,95 \text{ (тенге)}$$

Исходя из всех расчетов, полные расходы на электроэнергию

составляют:

$$Э = 190{,}95{,}95 + 3808 = 3998{,}95 \text{ (тенге)}$$

6.4 Расчет затрат на оплату труда

Для разработки программного обеспечения, как указывалось ранее, необходимо 3 работника:

– руководитель проекта – управление рабочим временем, корректировка рабочих процессов, координация, изучение предметной области;

– разработчик – разработка ПО, тестирование и сопровождение.

– аппаратный криптограф шифрование и дешифрование сообщений

Сумму расходов на оплату труда можно рассчитать по следующей формуле:

$$З_{тр} = \sum ЧС_i * T_i \quad (6.5)$$

где $ЧС_i$ – часовая ставка i–го работника, тг;

T_i – трудоемкость разработки модели, чел.×ч; i – категория работника;

n – количество работников, занятых разработкой ПП.

Во время реализации проекта рабочее время участников не равномерно, поэтому имеет смысл установить часовую ставку каждого работника и общий объем заработной платы.

Часовую ставку сотрудника можно рассчитать по следующей формуле:

$$ЧС_i = \frac{ЗП_i}{ФРВ_i} \quad (6.6)$$

где $ЗП_i$ – месячная заработная плата i–го работника, тг;

$ФРВ_i$ – месячный фонд рабочего времени i–го работника, час.

Месячная заработная плата руководителя равно 180 000 тенге и месячная заработная плата разработчика равно 150 000 тенге .Месячная зарплата аппаратного криптографа равно 160000 тенге . часовую ставку каждого работника согласно формуле (4.6):

$$ЧС_{Рук} = \frac{180\,000}{28 * 8} = 803{,}5 тг/ч$$

$$ЧС_{Разр} = \frac{150\,000}{28 * 8} = 699{,}6 тг/ч$$

$$ЧС_{Аппар} = \frac{160\,000}{28 * 8} = 714{,}2 \text{ тг/ч}$$

Часовая ставка руководителя составляет 803,5 (тг/ч), трудоемкость

разработки равно 100 часам. Часовая ставка разработчика составляет 699,6 (тг/ч), трудоемкость разработки равно 225 часам. Часовая ставка аппаратного криптографа равно 714,2(тг/ч) трудоемкость работы равно 120 часам . Согласно формуле (4.5) можно рассчитать сумму расходов на заработную плату работников:

$$З_{тр} = 803{,}5 * 100 + 699{,}6 * 225 + 714{,}2 * 120 = 80355 + 157470 + 85704 = 323524$$

Расчеты затрат по оплате труда показаны в таблице (6.5).

Таблица 4.5. – Расчет заработной платы

Категория работника	Квалификация	Трудоемкость разработки ПП, час.	Часовая ставка, тг/ч	Сумма, тг.
Руководитель	Проектный руководитель	100	803,5	80355
Разработчик	Программист	225	699,6	157470
Криптограф	Аппаратный	120	714,2	85704
Итого:				323524

6.5 Расчет затрат по социальному налогу

Согласно Налоговому кодексу Республики Казахстан социальный налог составляет 9,5% от фонта оплаты труда. Социальный налог можно рассчитать по следующей формуле:

$$С_н = (ФОТ - ПО) * 0{,}095 \quad (6.7)$$

где ПО – отчисления в пенсионный фонд, они составляют 10% от ФОТ.

$$ПО = 323524 * 0{,}1 = 32352{,}4 \text{ тенге}$$
$$С_н = (323524 - 32352{,}4) * 0{,}095 = 27661{,}3 \text{тенге}$$

Результаты расчетов представлены в таблице (6,6):

Таблица 6.6 – Начисление социального налога

Категория работника	Количество человек	Заработная плата, тг	Пенсионные отчисления, тг	Социальный налог, тг
Руководитель	1	80355	8 035	6 714,1

Разработчик	1	157470	15 577	13577
Криптограф	1	85704	8 996	5111,2
Итого:				25402,2

6.6 Амортизация основных фондов и прочие затраты

Нормы амортизации ОФ необходимо определить в соответствии с налоговым кодексом РК. Амортизацию ОФ можно определить по следующей формуле:

$$A_{\text{г}} = \frac{\text{С}_{\text{об}} * \text{Н}_{\text{а}}}{100} \quad (6.8)$$

где, $\text{С}_{\text{об}}$ – стоимость оборудования;

$\text{Н}_{\text{а}}$ – норма амортизации (норма амортизация = 25);

Формула (4.8) позволяет рассчитать нужную сумму для амортизационных отчислений за год для ноутбука:

$$A_{\text{г}} = \frac{390\,000 * 25}{100} = 97\,500 \text{ тенге}$$

Теперь необходимо рассчитать норму амортизации за период разработки:

$$A_{\text{г}} = \frac{97\,500 * 24}{365} = 6410{,}9 \text{ тенге}$$

Подобным образом необходимо рассчитать норму амортизации для всего оборудования. Результаты расчетов приведены в таблице (6.7).

Таблица 6.7 – Амортизация ОФ

Наименование оборудования и ПО	Стоимость оборудования и ПО, тг	Годовая норма амортизации, %	Сумма амортизации за год, тг	Сумма амортизации за время разработки, тг
Ноутбук	390 000	25	97 500	9 082,2
Принтер	52 874	25	13 218	108,64
Устройства шифрования	6700	20	1340	415,4
Модем	14000	20	2800	903
Итого:			114858,7	10509,52

Смета расходов на разработку ПО.

На основе всех представленных расчетов необходимо оформить смету

расходов на разработку ПО согласно форме, которая приведена в таблице (4.8).

Таблица 4.8 – Смета затрат на разработку ПО

Статьи затрат	Сумма, тг
Затраты на оборудование	444 038
Затраты на программное обеспечение	0
Затраты на оплату труда	323524
Социальные налоги	25402,2
Затраты на электроэнергию	3998,95
Амортизация основных фондов	10509,52
Прочие расходы	8560
Итого по смете:	816031,4

Рисунок 6.1 – Диаграмма затрат

6.7 Определение возможной (договорной) цены ПО

Стоимость $Ц_Д$ для программного обеспечения можно рассчитать по следующей формуле:

$$Ц_Д = З_{нир}\left(1 + \frac{P}{100}\right), \quad (6.9)$$

где $З_{нир}$ – затраты на разработку программного обеспечения, тг;

P – средний уровень рентабельности ПО, (%). Данный параметр принят равным 25%.

$$Ц_Д = 816031{,}4\left(1 + \frac{25}{100}\right) = 816031{,}4 + 204007{,}85 = 1020039{,}25 \text{ тенге}$$

Далее необходимо определить стоимость реализации с учетом НДС, ставка НДС устанавливается законодательством РК. На 2019 года ставка НДС составляет 12%. Стоимость реализации учитывая НДС можно рассчитать по следующей формуле:

$$Ц_р = Ц_д + Ц_д * НДС, \quad (6.10)$$

$$Ц_р = 1020039{,}25 + 1020039{,}25 * 0{,}12 = 1020039{,}25 + 122404{,}71 = 1142443{,}96 \text{ тенге}$$

Рассчет показал что договорная цена ПП равно 1142443,96тг что является в приделах нормы.Расчеты показали что себестоимость была равна 816031,40 тг. Прибыль же составила 204007,85 тг.

Заключение

В данной дипломным проекте мой способ аппаратного шифрования позволяет существенно снизить нагрузку на аппаратную систему и тем самым повысить эффективность шифрования. Был проведен анализ существующих способов формирования остатков от числа по модулю. При этом количественно оценены различные способы формирования остатков по модулю: по времени формирования остатков, по сложности их схемной реализации. Определен предпочтительный способ формирования остатка по модулю–формирование остатка на основе делительного устройства. Также не стоит забывать мы произвели конверезацию транзакций. В разделе БЖД мы выяснили, что людям не хватает естественного освещения в помещении;в технико–экономическом обосновании мы подсчитали, что все затраты оборудование в принципе входят экономическую норму.

Список сокращений

ФЧО – Формирователь частичных остатков
СС - Схема сравнения
ЧО - Частичный остаток
Рга – Регистр
СМ - Сумматор
Л.3 – Линия задержки
ТИ - Тактовый импульс
СЧ - Счетчик

Список литературы

1 Рябко Б. Я., Фионов А. Н. Основы современной криптографии для специалистов в информационных технологиях. –М.: Научный мир, 2004.

2 Е. Ж. Айтхожаева, С. Г. Тынымбаев Аспекты аппаратного приведения по модулю в ассиметричной криптографии // –Алматы: Вестник НАН РК №5. 2014.

3 Орлов С. А., Цилькер Б. Я. Организация ЭВМ и систем. 3–е издание. – СПб.: Питер, 2014.

4 Цилькер Б. Я., Орлов С. А. Организация ЭВМ и систем. 2–е изд. СПб.: Питер, 2011.

5 Калашников О. Ассемблер – это просто. Учимся программировать. – М.: Научный мир 2011 г.

6 Алгоритм шифрования RSA URL http://enisey.name/umk/pzis/ch18s07.html (дата обращения 15.05.2019)

7 Ковтун В.Ю.Охрименко А.А. Умножения целых чисел с использованием отложеного переноса для криптосистем с открытым ключом. – Информационные технологи и системы в управлении, образовании, науке: Монография / Под ред.. проф. В.С. Пономаренко. – М: Цифрова друкарня №1, 2013.– С. 69–82. – ISBN978–617–7017–37–9.

8 Рябко Б.Я, Фионов А.И «Основы ассиметричной криптографии». М.: Научный мир, 2004.

9 Свидетельство № 1621 от 3 июля 2017 г., ИС 009183 о гос. регистрации прав на объект авторского права «Алгоритм деления чисел и устройство для его осуществления». В соавторстве с Тынымбаев С., Жайбергенова Ж.А., Иманбаев А.Ж., Зиро А.А.

10 Бекишева А.И. Методические указания к выполнению экономической части дипломной работы для бакалавров специальности 5В0703 – Информационные системы – Алматы: АУЭС, 2013

Приложение А

Листинг кода

```
.model tiny
.data
A       dw      48753
B       dw      32298
A1      dd      00h
B1      dd      00h
AB      dd      00h
.code
org 100h
start:
        mov     ax,A
        mov     dx,ax
        shl     ax,02h
        shr     dx,0Eh
        cmp     dx,02h
        jc      nW1
        not     ax
        not     dx
        add     ax,01h
        adc     dx,00h
nW1:    mov     word ptr A1,ax
        mov     word ptr A1[02h],dx
        mov     ax,B
        mov     dx,ax
        shl     ax,03h
        shr     dx,0Dh
        cmp     dx,04
        jc      nW2
        not     ax
        not     dx
        add     ax,01h
        adc     dx,00h
nW2:    mov     word ptr B1,ax
        mov     word ptr B1[02h],dx
        mov     bx,word ptr A1
        mov     cx,word ptr A1[02h]
        add     ax,word ptr A1
        adc     dx,word ptr A1[02h]
ret                             ;
end start
```