**МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ, СВЯЗИ**

**И МАССОВЫХ КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ**

Ордена Трудового Красного Знамени федеральное государственное бюджетное образовательное учреждение высшего образования

**Московский технический университет связи и информатики**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Кафедра многоканальных телекоммуникационных систем

**КУРСОВОЙ ПРОЕКТ**

по дисциплине

**Цифровые системы передачи**

**и методы их защиты**

на тему

**РЕАЛИЗАЦИЯ НА ОСНОВЕ СИГНАЛЬНОГО ПРОЦЕССОРА КРИПТОСИСТЕМЫ ПО АЛГОРИТМУ RSA**

**В РАСШИРЕННЫХ ПОЛЯХ ГАЛУА. ШИФРОВАНИЕ 48-БИТНОЕ**

Бригада №9

Выполнил:

студент гр. БЗС2101

Рожнов М.С.

Проверила:

ст. преп. Зуйкова Т.Н.

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 3](#_Toc185925750)

[ГЛАВА 1. ФОРМИРОВАНИЕ ТЕХНИЧЕСКОГО ЗАДАНИЯ НА СОСТАВНУЮ ЧАСТЬ ЭСКИЗНОГО ПРОЕКТА 4](#_Toc185925751)

[ГЛАВА 2. КРАТКИЕ ТЕОРЕТИЧЕСКИЕ ОСНОВЫ 6](#_Toc185925752)

[ГЛАВА 3. РАЗРАБОТКА ФУНКЦИОНАЛЬНОЙ СХЕМЫ МИКРОПРОЦЕССОРНОЙ КРИПТОГРАФИЧЕСКОЙ СИСТЕМЫ 7](#_Toc185925753)

[ГЛАВА 4. РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ МИКРОПРОЦЕССОРНОГО УСТРОЙСТВА 10](#_Toc185925754)

[4.1 Разработка блок-схемы генерации ключей 10](#_Toc185925755)

[4.3 Разработка блок-схем подпрограмм основной программы 13](#_Toc185925756)

[ГЛАВА 5. АНАЛИЗ РЕЗУЛЬТАТОВ РАБОТЫ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ МИКРОПРОЦЕССОРНОГО УСТРОЙСТВА 24](#_Toc185925757)

[ЗАКЛЮЧЕНИЕ 27](#_Toc185925758)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 28](#_Toc185925759)

[ПРИЛОЖЕНИЕ А 29](#_Toc185925760)

[ПРИЛОЖЕНИЕ Б 38](#_Toc185925761)

[ПРИЛОЖЕНИЕ В 45](#_Toc185925762)

[ПРИЛОЖЕНИЕ Г 47](#_Toc185925763)

# ВВЕДЕНИЕ

В данном курсовом проекте разработана микропроцессорная криптографическая система для защиты от несанкционированного доступа информации, передаваемой в первичном цифровом потоке E1, на основе сигнального процессора ADSP-2181 по алгоритму RSA в расширенных полях Галуа. В частности, система будет как шифровать, так и расшифровывать.

Говоря об актуальности данной темы, защита информации была и будет востребована; всегда будет нужда в конфиденциальности данных и их защиты от злоумышленников, которые вместе с технологиями защиты информации развивают свои методы взлома криптосистем. Поэтому занимаясь данной работой получен опыт в области криптозащиты информации и проектирования криптосистем.

В курсовом проекте реализован алгоритм RSA в расширенных полях Галуа с разрядной сеткой 48 бит. Это криптографический алгоритм с открытым ключом, основывающийся на неравных больших взаимно простых числах. 48 бит разрядная сетка позволила увеличить количество обрабатываемой информации и степень защищенности из-за большего полинома.

Целью курсового проектирования является усвоение теоретических знаний в области защиты информации, а также приобретение практических навыков криптографических систем. Перед началом практической части проекта, было проведена работа с теоретической частью, а именно алгоритмом RSA в расширенных полях Галуа в структуре потока E1, и согласовать техническое задание с преподавателем. В практической части спроектирована функциональная схема микропроцессорного устройства и написана программа для шифрования сообщения.

# ГЛАВА 1. ФОРМИРОВАНИЕ ТЕХНИЧЕСКОГО ЗАДАНИЯ НА СОСТАВНУЮ ЧАСТЬ ЭСКИЗНОГО ПРОЕКТА

Микропроцессорная криптографическая система   
наименование, шифр

шифрования/расшифрования по алгоритму RSA   
в расширенных полях Галуа МПКС-RSA-ШР-n-2024-09.

Микропроцессорное криптографическое устройство   
обозначение составной части изделия  
для шифрования МПКС-RSA-ШР-n-2024-09(Ш)

Техническое задание было сформировано поэтапно и описано в приложении А. Сначала было согласовано наименование эскизного проекта на основе темы курсовой работы, и чтобы не использовать громоздкое название, был сформулирован шифр.

Цель разработки и технические требования к изделью основываются на задании на курсовой проект. Они обсуждены и согласованы с преподавателем Зуйковой Татьяной Николаевной. На основе задания сформировано техническое задание, в котором требуется запрограммировать последовательный порт SPORT0, программы работы в расширенных полях Галуа и шифрование/расшифрование алгоритмом RSA.

Так как посчитать функцию Эйлера в расширенных полях Галуа достаточно проблематично, то для данного курсового проекта в качестве открытого ключа **n** был взят неприводимый полином [4], для которого функция Эйлера считается просто. В качестве ключа **e** было взято взаимно простое функции Эйлеру число [7036874419]. Взаимно обратное ему было найдено по методу расширенного алгоритма Евклида – [44142312652261]. Для расчетов и проверок использовался язык программирования Python. До начала работы программы ключи заносятся в нее вручную и остаются неизменными весь период работы.

В макете программы использовались порты IO(0), как входной, и IO(1), как выходной.

Совместно со студентом Кожевниковой Д.Е. была разработана концепция микропроцессорной системы и эскиз проекта. Самостоятельно была разработана программа шифрования алгоритма RSA в расширенных полях Галуа в 48 разрядной сетке.

# ГЛАВА 2. КРАТКИЕ ТЕОРЕТИЧЕСКИЕ ОСНОВЫ

Алгоритм RSA изобретен в 1978 году Роном Ривестом, Ади Шамиров и Леонардом Адлеманом. Криптоустойчивость алгоритма заключается в сложности разложения больших числе на простые сомножители.

Для реализации криптосистемы, функционирующей в соответствии с алгоритмом RSA, необходимо выполнить следующие действия:

- выбрать два неравных больших простых числа **p** и **q.**

- вычислить из произведение

- выбрать (произвольное) число **e**, которое должно быть взаимно простым (не должно содержать общих сомножителей) с и .

- найти число **d**, обратное **e** по модулю , т.е. , или

Числа **e** и **n** составляют открытый ключ, а число **d** является закрытым (индивидуальным) ключом алгоритма. Числа **p** и **q** не используются.[1]

Далее каждый блок шифруется в соответствии с выражением:

Блоки являются блоками зашифрованного сообщения.

Дешифрование блоков производится в соответствии с выражением:

В расширенных полях Галуа операции сложения и вычитания реализуются логической операцией «исключающее ИЛИ», не приводят к расширению разрядной сетки и, если операнды являются элементами поля, то не требуют применения операции вычисления остатка от деления на образующий полином по отношению к результату операции. Однако остальные операции, такие как умножение и деление, реализуются несколько сложнее.[2]

# ГЛАВА 3. РАЗРАБОТКА ФУНКЦИОНАЛЬНОЙ СХЕМЫ МИКРОПРОЦЕССОРНОЙ КРИПТОГРАФИЧЕСКОЙ СИСТЕМЫ

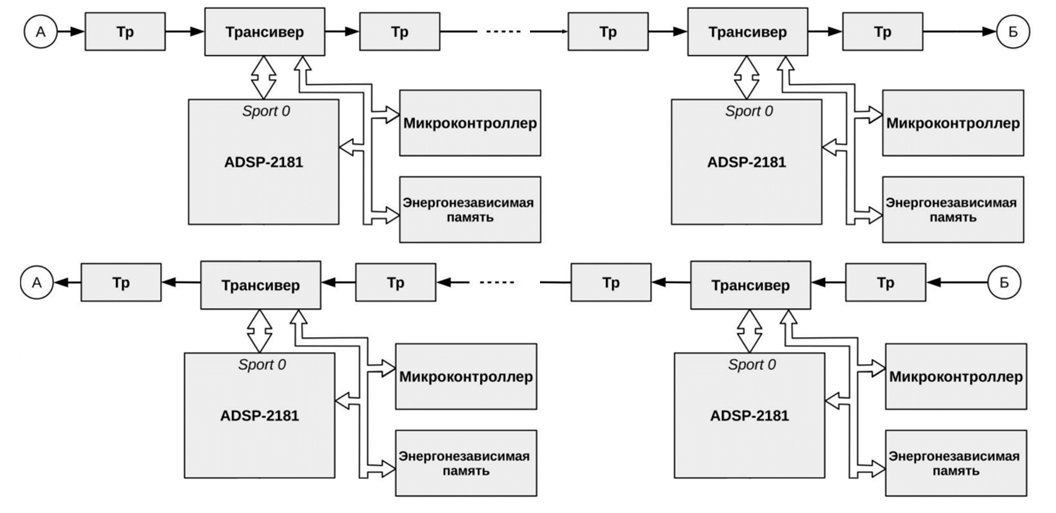
На рисунке 2 представлена функциональная схема микропроцессорной криптосистемы абонента А на базе сигнального процессора ADSP-2181.

Рисунок 3.1 - Функциональная схема микропроцессорной криптосистемы двух направлений передачи, включенной в поток Е1

В микропроцессорной криптосистеме выполнение программ осуществляет сигнальный процессор, а непосредственно программное обеспечение записано в энергонезависимой памяти. Такое решение продиктовано тем, что при каждом выключении питания память программ процессора очищается, и в этом случае при каждом рестарте пришлось бы заново программировать процессор. По рестарту программа автоматически считывается из ячеек энергонезависимой памяти во внутреннее ОЗУ программ процессора, после чего активизируется процесс её выполнения. Взаимодействия с энергонезависимой памятью осуществляется через микроконтроллер.

Трансивер первичного цифрового потока представляет собой специализированный микропроцессор, ориентированный на работу с цифровым потоком E1, поступающим на его вход. Входное сопротивление каждого трансивера согласуется с сопротивлением линии посредством включения трансформаторов Тр. Параметры работы трансивера (код сигнала, структура цифрового потока и т.п.) управляются микроконтроллером.

Два последовательных порта SPORT0 и SPORT1, функциональная схема которых приведена на рисунке 3, предназначены для обмена информацией с внешними устройствами в последовательном режиме. Последовательный обмен предполагает вывод информации по одному проводу и одновременно ввод по другому проводу последовательно бит за битом на заранее запрограммированной скорости или под воздействием сигнала внешнего генератора. Передача/прием информации последовательными портами процессора ADSP-2181 осуществляется начиная со старшего разряда (в отличие от последовательных портов микроконтроллеров, ориентированных на обеспечение обмена информацией с компьютерами).

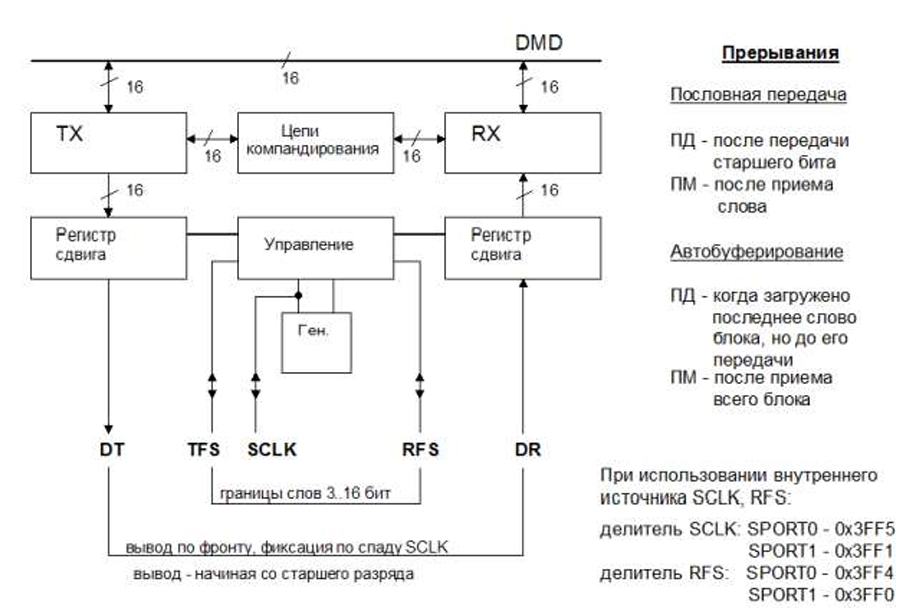


Рисунок 3.2 - Функциональная схема (архитектура) последовательных портов

Процессор ADSP-2181 представляет собой один из наиболее мощных 16 - разрядных процессоров семейства ADSP -21XX, производимых фирмой Analog Devices. Процессор построен по оригинальной пятишинной архитектуре, оптимизированной с ориентацией на обработку сигналов в формате с фиксированной точкой.

Типовая функциональная схема, иллюстрирующая включение последовательного порта процессора ADSP-2181 в разрыв одного из направлений передачи потока Е1 с использованием приемопередатчика (ПП) представлена на рисунке 4.

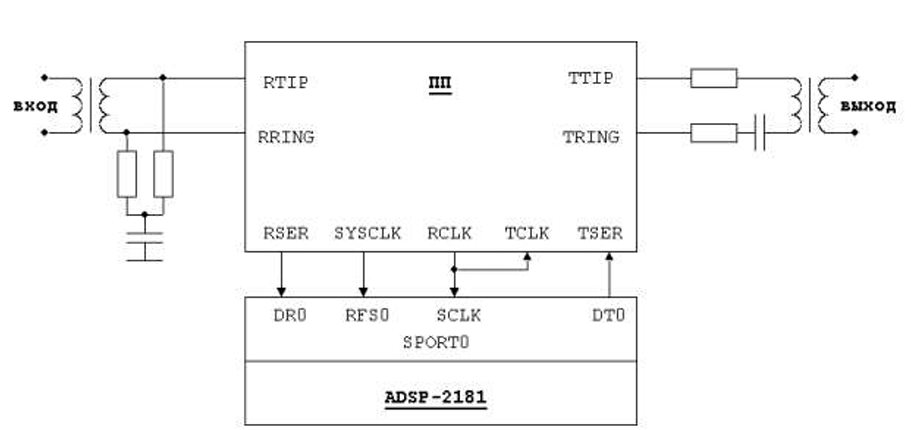


Рисунок 3.3 - Функциональная схема включения микропроцессора в поток Е1

На этом рисунке обозначены следующие выводы приемопередатчика ПП:

RTIP, RRING – вход по потоку Е1;

TTIP, TRING - выход по потоку Е1;

RSER – выход принимаемого сигнала в коде БВН(NRZ);

TSER – вход передаваемого сигнала в коде БВН(NRZ);

RCLK – выход тактового синхросигнала тракта приема;

TCLK – вход тактового синхросигнала тракта передачи;

SYSCLK - выход циклового синхросигнала.

# ГЛАВА 4. РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ МИКРОПРОЦЕССОРНОГО УСТРОЙСТВА

## 4.1 Разработка блок-схемы генерации ключей

Генерация ключей производится вне состава криптографической системы.

Для генерации пары ключей (открытого и закрытого) в алгоритме RSA необходимо выбрать большие (>1050) простые числа **p** и **q** и вычислить их произведение ***n****=p\*q*, однако, как было в главе 1, для упрощения выполнения работы, в качестве **n** был использован неприводимый полином. Также необходимо выбрать число е, взаимно простое с функцией Эйлера **phi(n)** и сгенерировать взаимно обратное **d**.

Генерация ключей производится посредством языка Python и приведена в приложении

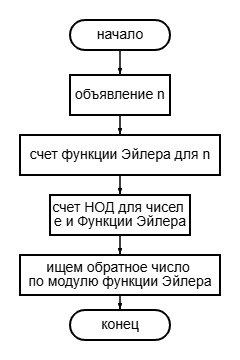


Рисунок 4.1.1 - Блок-схема генерации ключей алгоритма RSA в расширенных полях Галуа 48-битовых

**4.2 Разработка блок-схемы алгоритма основной программы**

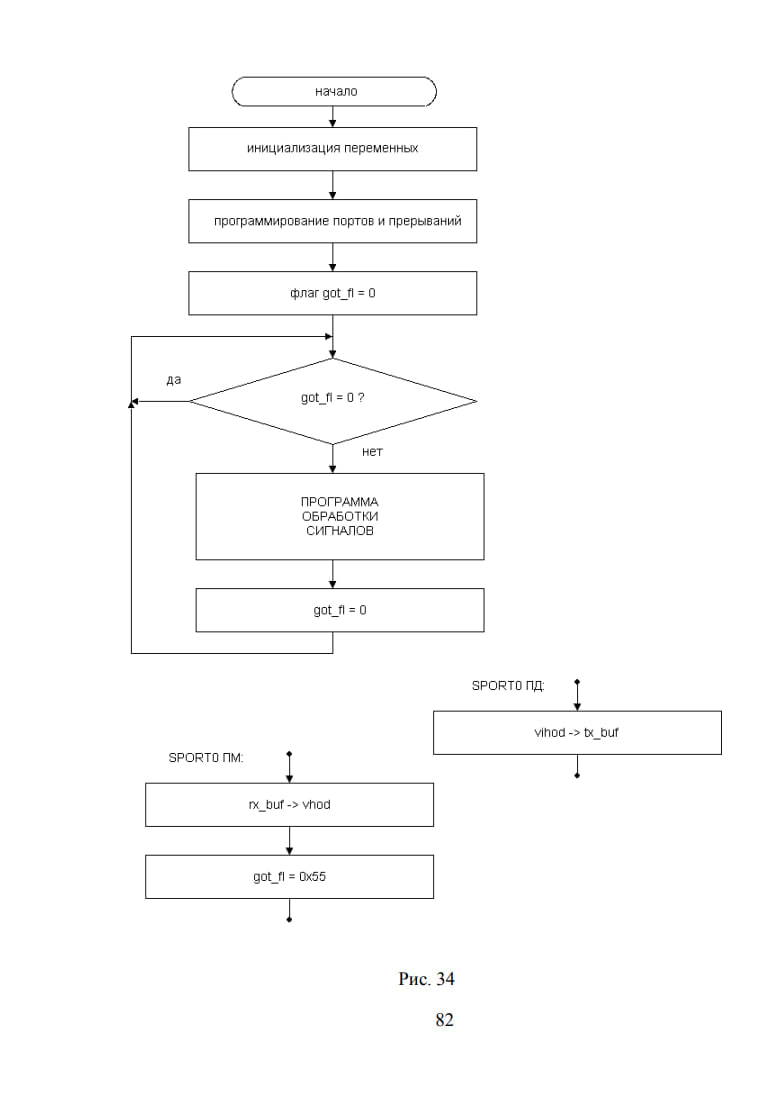


Рисунок 4.2.1 - Блок-схема программирования последовательного порта SPORT0

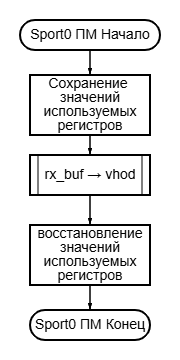
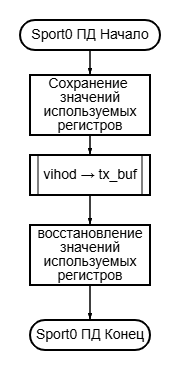


Рисунок 4.2.2 - Блок-схема прерывающих подпрограмм SPORT0 для приёма и передачи

По окончании очередного цикла портом сначала формируется запрос на прерывание “передатчик пуст”. По этому запросу прерывающая подпрограмма должна немедленно переписать результаты обработки, сформированные в буфере Vihod на предшествующем цикле, в буфер tx\_buf.

Спустя 3,9 мкс порт формирует запрос на прерывание “приёмник полон”

Соответствующая программа должна немедленно переписать содержимое буфера rx\_buf в буфер vhod.

## 4.3 Разработка блок-схем подпрограмм основной программы

Помимо блоков инициализации и программирования портов и прерываний, основная программа также содержит блок программы обработки сигнала, который состоит из подпрограмм, блок-схемы которых будут приведены ниже.

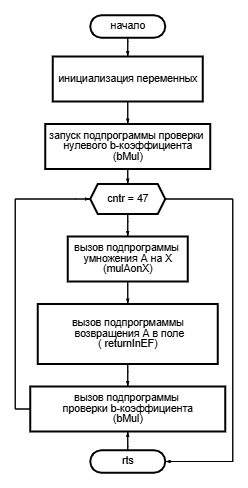


Рисунок 4.3.1 – Основная блок-схема умножения 48-битового числа в расширенных полях

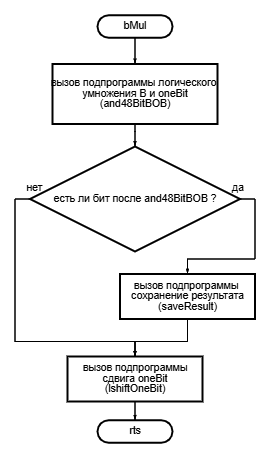


Рисунок 4.3.2 - Блок-схема подпрограммы проверки бита во втором множителе (bMul)

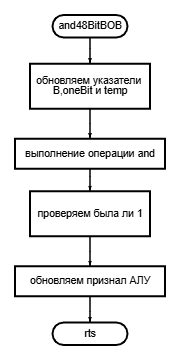


Рисунок 4.3.3 - Блок-схема подпрограммы операции логического умножения 48-битных чисел (and48BitBOB)

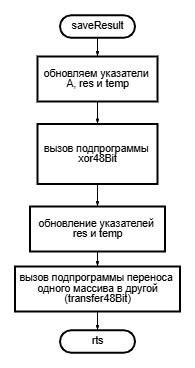


Рисунок 4.3.4 - Блок-схема подпрограммы сохранения результата xor-операций через массив temp (saveResult)

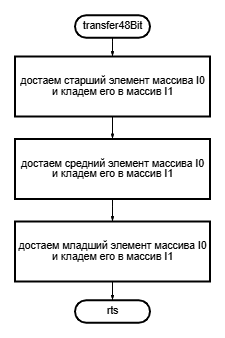


Рисунок 4.3.5 - Блок-схема подпрограммы переноса значений массива I0 в массив I1 (transfer48Bit)

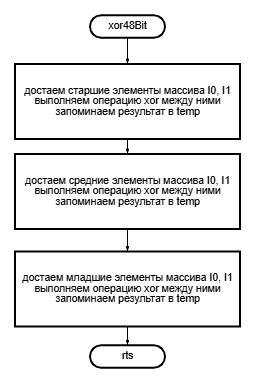


Рисунок 4.3.6 - Блок-схема подпрограммы операции xor между элементами массива (xor48Bit)

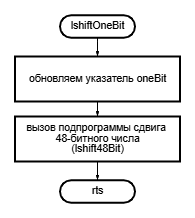


Рисунок 4.3.7 - Блок-схема подпрограммы сдвига влево на один бит массива oneBit (lshiftOneBit)



Рисунок 4.3.8 - Блок-схема подпрограммы сдвига элементов массива (lshift48Bit)

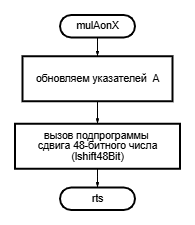


Рисунок 4.3.9 - Блок-схема подпрограммы умножения A на X (mulAonX)



Рисунок 4.3.10 - Блок-схема подпрограммы возвращение A в поле (returnInEF)

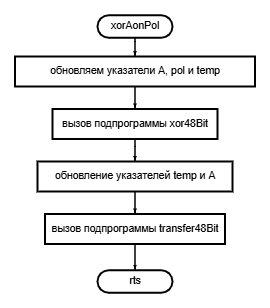


Рисунок 4.3.11 - Блок-схема подпрограммы xor между A и Pol и перенос массива в другой (xorAonPol)

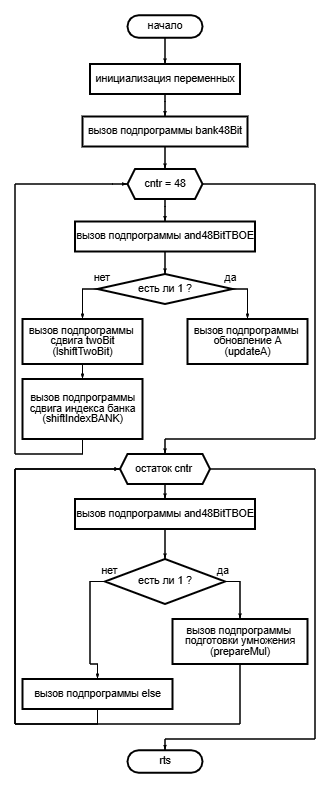


Рисунок 4.3.12 – Основная блок-схема программы возведения в степень 48-битного числа в расширенных полях

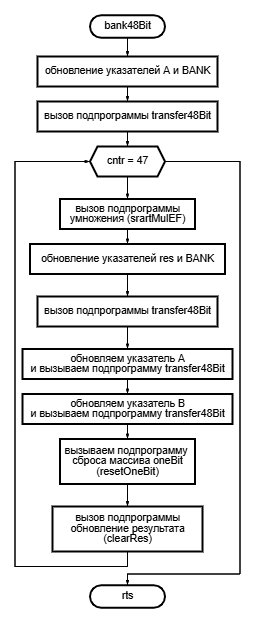


Рисунок 4.3.13 - Блок-схема подпрограммы формирования таблицы степеней квадратов (bank48Bit)

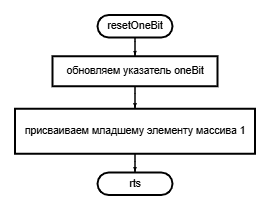


Рисунок 4.3.14 - Блок-схема подпрограммы сброса массива oneBit (resetOneBit)



Рисунок 4.3.15 - Блок-схема подпрограммы обнуление ячеек массива res (clearRes)

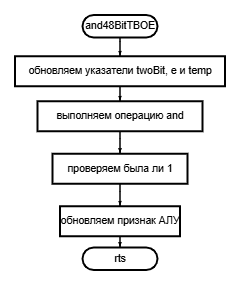


Рисунок 4.3.16 - Блок-схема подпрограммы операции логического умножения массива TBOE (and48BitTBOE)

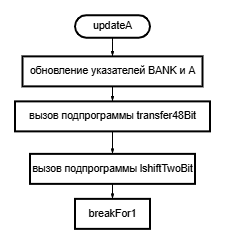


Рисунок 4.3.17 - Блок-схема подпрограммы обновление массива A переменной из массива BANK (updateA)

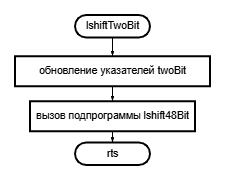


Рисунок 4.3.18 - Блок-схема подпрограммы сдвига twoBit (lshiftTwoBit)



Рисунок 4.3.19 - Блок-схема подпрограммы сдвига индекса банка (shiftIndexBANK)

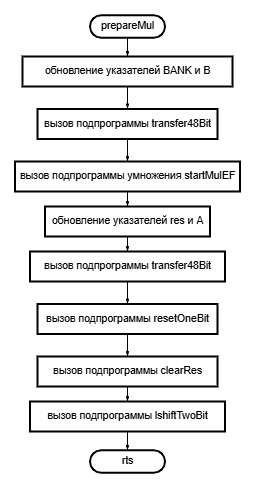


Рисунок 4.3.20 - Блок-схема подпрограммы подготовки умножения (prepareMul)

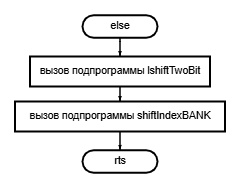


Рисунок 4.3.21 - Блок-схема подпрограммы ответвления от условия в цикле блока подпрограммы возведения в степень (else)

# ГЛАВА 5. АНАЛИЗ РЕЗУЛЬТАТОВ РАБОТЫ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ МИКРОПРОЦЕССОРНОГО УСТРОЙСТВА

Результат работы программы на Python по подсчету ключей RSA на рисунке 5.1:



Рисунок 5.1 – Вывод программы счета ключей

Результат отладки подпрограммы умножения 48 битовых чисел в расширенных полях Галуа. В качестве доказательства правильной работы подпрограммы, были взяты входные данные на рисунке 5.2 и запущенны в среде ADSP-2181 и Python для сравнения результатов:

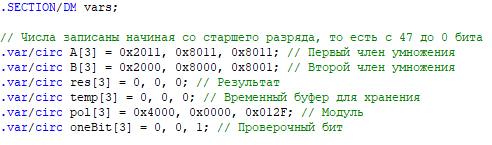


Рисунок 5.2 – Входные данные подпрограммы умножения



Рисунок 5.3 – Результат выполнения подпрограммы на ADSP-2181

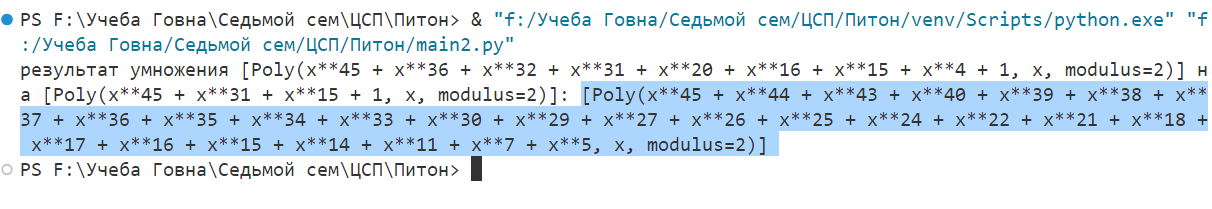


Рисунок 5.4 – Результат выполнения подпрограммы на Python

Результат отладки подпрограммы возведения в степень 48 битовых чисел в расширенных полях Галуа. В качестве доказательства правильной работы подпрограммы, были взяты входные данные на рисунке 5.5 и запущенны в среде ADSP-2181 и Python для сравнения результатов:

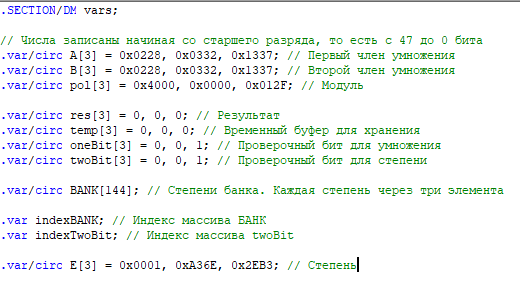


Рисунок 5.2 – Входные данные подпрограммы возведения в степень



Рисунок 5.3 – Результат выполнения подпрограммы на ADSP-2181

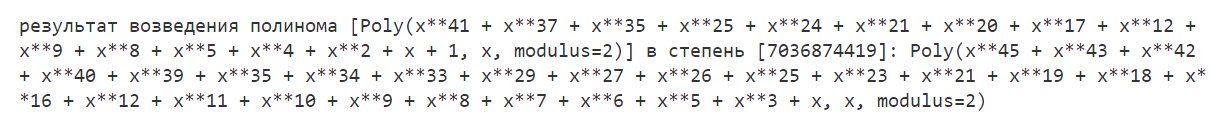


Рисунок 5.4 – Результат выполнения подпрограммы на Python

Результаты основной программы шифрования 48 битовых чисел в расширенных полях Галуа представлены на рисунке 5.5 и 5.6:

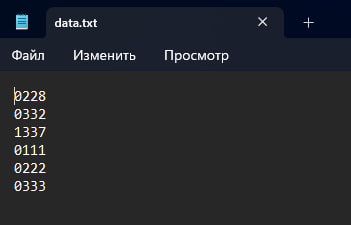


Рисунок 5.5 – Входные данные в файле

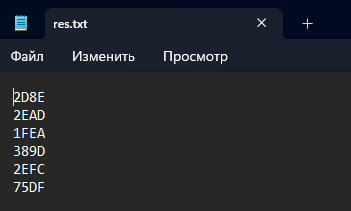


Рисунок 5.6 – Файл с результатом работы программы

По текущим результатам работы программы и подпрограмм можно сделать вывод, что подпрограммы умножения и возведения в степень работают правильно и шифрование происходит корректно.

# ЗАКЛЮЧЕНИЕ

В данном курсовом проекте был разработан алгоритм шифрования RSA на языке Ассемблер для процессора ADSP-2181. Для реализации задачи были рассмотрены математическая модель формирования, составлена блок-схема, написана и отлажена программа. После выполнения программы, конечные значения совпали со значениями, полученными в среде Python, следовательно, программа выполнена верно.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Шаврин С.С. Защита информации в многоканальных телекоммуникационных системах. Часть 1: учебное пособие/С.С. Шаврин. - М.: МТУСИ, 2002. - 62 с.
2. Реализация базовых операций защиты информации на сигнальных процессах. Часть 3: учебное пособие/С.С. Шаврин, Т.Н. Зуйкова, О.Ю. Мусатова. - М.: МТУСИ, 2018. - 36 с.
3. Реализация базовых операций защиты информации на сигнальных процессах. Часть 2: учебное пособие/С.С. Шаврин. - М.: МТУСИ, 2016. - 42 с.
4. Б. Шнайер, Прикладная криптография. Протоколы, алгоритмы, исходные тексты на языке С/Б. Шнайер – М.: Изд-во Триумф, 2002. - 816 стр.
5. Технологии микропроцессорных систем в инфокоммуникациях: учебное пособие/С.С. Шаврин, С.В. Мельник. - М.: МТУСИ, 2020. - 92 с.
6. Реализация генераторов ключевых последовательностей на сигнальных процессорах. Практикум. Ч. 1: учебное пособие/С.В. Мельник, С.С. Шаврин, Т.Н. Зуйкова, О.Ю. Мусатова. М.: МТУСИ, 2017. - 30 с.
7. Реализация генераторов ключевых последовательностей на сигнальных процессорах. Практикум. Часть 2: учебное пособие/С.В. Мельник, С.С. Шаврин, Т.Н. Зуйкова, О.Ю. Мусатова. М.: МТУСИ, 2018. - 30 с.
8. Учебное-методическое пособие по курсовому проектированию по дисциплине «Цифровые системы передачи и методы их защиты»/Т.Н. Зуйкова, О.Ю. Мусатова. М.: МТУСИ, 2023. - 28 с.
9. Возведение в степень в простых полях Галуа. Практикум: учебное пособие/С.С. Шаврин, О.Ю. Мусатова. М.: МТУСИ, 2024. - 28 с.

# ПРИЛОЖЕНИЕ А

|  |  |  |
| --- | --- | --- |
| *СОГЛАСОВАНО*  ст. преп. Зуйкова Т.Н.  должность, заказчик  Т.Н. Зуйкова подпись, инициалы, фамилия  «\_\_\_\_»\_\_\_\_\_\_\_\_\_20\_\_\_ г. | *СОГЛАСОВАНО*  студент гр. БЗС2101  должность, исполнитель составной части проекта  М.С. Рожнов подпись, инициалы, фамилия  «\_\_\_\_»\_\_\_\_\_\_\_\_\_20\_\_\_ г. | *УТВЕРЖДАЮ*  руководитель КП  должность, руководитель проекта  Т.Н. Зуйкова подпись, инициалы, фамилия  «\_\_\_\_»\_\_\_\_\_\_\_\_\_20\_\_\_ г. |
| *СОГЛАСОВАНО*  студент гр. БЗС2101  должность, исполнитель составной части проекта  Д.Е. Кожевникова подпись, инициалы, фамилия  «\_\_\_\_»\_\_\_\_\_\_\_\_\_20\_\_\_ г. | *СОГЛАСОВАНО*  студент гр. БЗС2101  должность, исполнитель составной части проекта  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ подпись, инициалы, фамилия  «\_\_\_\_»\_\_\_\_\_\_\_\_\_20\_\_\_ г. |  |
| ТЕХНИЧЕСКОЕ ЗАДАНИЕ  НА СОСТАВНУЮ ЧАСТЬ ЭСКИЗНОГО ПРОЕКТА  Микропроцессорная криптографическая система  наименование, шифр  шифрования/расшифрования по алгоритму RSA  в расширенных полях Галуа МПКС-RSA-ШР-n-2024-09.  Микропроцессорное криптографическое устройство  обозначение составной части изделия  для шифрования МПКС-RSA-ШР-n-2024-09(Ш) | | |

**1. Наименование, шифр, основание, исполнитель и сроки выполнения эскизного проекта**

1.1 Наименование эскизного проекта - Микропроцессорная криптографическая система шифрования/расшифрования по алгоритму RSA в расширенных полях Галуа

1.2 Шифр - МПКС-RSA-ШР-n-2024-09

1.3 Основание

1.3.1 Рабочая программа дисциплины Б1.В.22 «Цифровые системы передачи и методы их защиты». Актуализирована на 2024/25 уч.г.

1.3.2 Учебный план по программе бакалавриата «11.03.02 Инфокоммуникационные технологии и системы связи. Защищенные инфокоммуникационные системы». Одобрен Ученым советом МТУСИ 30.08.2021.

1.3.3 Задание на курсовое проектирование по дисциплине «Цифровые системы передачи и методы их защиты»: тема КП «Реализация на основе сигнального процессора криптосистемы по алгоритму RSA в расширенных полях Галуа. Шифрование 48-битное»

1.3.4. Учебно-методическое пособие по курсовому проектированию по дисциплине «Цифровые системы передачи и методы их защиты» / Авторы-составители Т.Н. Зуйкова, О.Ю. Мусатова. – М.: МТУСИ, 2023. – 32 с.

1.4 Исполнители

Исполнитель составной части МПКС-RSA-ШР-n-2024-09(Ш) - студент 4 курса бакалавриата основной профессиональной образовательной программы «Инфокоммуникационные технологии и системы связи. Защищенные инфокоммуникационные системы» М.С. Рожнов

инициалы, фамилия исполнителя

Исполнитель составной части МПКС-RSA-ШР-n-2024-09(Р) - студент   
4 курса бакалавриата основной профессиональной образовательной программы «Инфокоммуникационные технологии и системы связи. Защищенные инфокоммуникационные системы» Д.Е. Кожевникова

инициалы, фамилия исполнителя

1.5 Сроки выполнения

Подготовительный этап — с 2-й по 6-ю недели 7 семестра

Проектный этап — с 6-й по 10-ю недели 7 семестра

Реализационный этап — с 10-й по 14-ю недели 7 семестра

Оформительский этап — с 6-й по 15-ю недели 7 семестра

Заключительный этап — 15-17 недели 7 семестра.

**2 Цель разработки, наименование и обозначение изделия**

**2.1 Цель разработки**

В результате выполнения составной части эскизного проекта должно быть разработано микропроцессорное криптографическое устройство для шифрования в составе микропроцессорной криптографической системы шифрования/расшифрования по алгоритму RSA в расширенных полях Галуа.

2.2 Наименование изделия — микропроцессорное криптографическое устройство для шифрования.

2.3 Обозначение изделия — МПКС-RSA-ШР-n-2024-09(Ш).

2.4 Основное назначение — шифрование информации, передаваемой по отдельному канальному интервалу в первичном цифровом потоке Е1.

**3 Технические требования к изделию**

3.1 Состав изделия

В состав микропроцессорной криптографической системы шифрования/расшифрования по алгоритму RSA в расширенных полях Галуа входят две составные части:

- микропроцессорное криптографическое устройство для шифрования;

- микропроцессорное криптографическое устройство для расшифрования.

3.2. Требования назначения

3.2.1 Алгоритм шифрования — RSA.

3.2.2 Математический аппарат реализации алгоритма — расширенные поля Галуа.

3.2.3 Разрядность обрабатываемых данных — 48.

3.2.4 Данные для шифрования поступают из канального интервала первичного цифрового потока Е1 — КИ09.

3.2.5 Данные поступают через последовательный порт SPORT0.

3.2.6 Для взаимодействия с микропроцессорным криптографическим устройством для расшифрования в тракте приема используется последовательный порт SPORT0.

3.3 Конструктивные требования

Не предусмотрены.

3.4 Требования электромагнитной совместимости

Не предусмотрены.

3.5 Требования живучести и стойкости к внешним воздействия

Не предусмотрены.

3.6 Требования надежности

Не предусмотрены.

3.7 Требования эргономики, обитаемости и технической эстетики

Не предусмотрены.

3.8 Требования к эксплуатации, хранению, удобству технического обслуживания и ремонта

Не предусмотрены.

3.9 Транспортирование

Не предусмотрены.

3.10 Требования безопасности

Не предусмотрены.

3.11 Требования стандартизации, унификации и каталогизации

Не предусмотрены.

3.12 Требования технологичности

Не предусмотрены.

**4 Технико-экономические требования**

Не предусмотрены.

**5 Требования к видам обеспечения**

5.1 Требования к нормативно-техническому обеспечению

«Об информации, информационных технологиях и защите информации»  
 «О безопасности» (Федеральный закон от 28.12.2010 г., N 390-03)  
Рекомендации МСЭ-Т  
ГОСТ Р 34.10-2012 «Информационная технология  
Криптографическая защита информации»

5.2 Требования к метрологическому обеспечению

Не предусмотрены.

5.3 Требования к диагностическому обеспечению

Не предусмотрены.

5.4 Требования к математическому, программному и информационно-лингвистическому обеспечению

- пакет программ, симулирующих функции сигнального процессора ADSP-2181 компании AnalogDevices

- среда разработки и отладки программ VisualDSP++

**6 Требования к сырью, материалам и КИПМ**

Компоненты: микросхемы микроконтроллера, трансивер первичного цифрового потока, процессор ADSP-2181, 16-разрядная память данных

**7 Требования к консервации, упаковке и маркировке**

Отчет предоставляется в бумажном виде и электронном формате (DOC и PDF).

Рабочие файлы в формате asm и py.

**8 Требования к учебно-тренировочным средствам**

Не предусмотрены.

**9 Специальные требования**

Не предусмотрены.

**10 Требования к документации**

10.1 Перечень документов, поставляющихся с программным обеспечением микропроцессорного криптографического устройства для расшифрования:

10.1.1 Пояснительная записка курсового проекта

10.1.2 Схема функциональная МПКС

10.1.3 Программное обеспечение микропроцессорного криптографического устройства для расшифрования

10.1.4 Программное обеспечение демонстрационной версии МПКС

10.1.5 Инструкция по применению МПКС

10.2 Документы предоставляются в бумажном виде и электронном формате (DOC и PDF).

**11 Этапы выполнения эскизного проекта**

Подготовительный этап — с 2-й по 6-ю недели 7 семестра

Проектный этап — с 6-й по 10-ю недели 7 семестра

Реализационный этап — с 10-й по 14-ю недели 7 семестра

Оформительский этап — с 6-й по 15-ю недели 7 семестра

Заключительный этап — 15-17 недели 7 семестра.

**12 Порядок выполнения и приемки этапов эскизного проекта**

|  |  |  |
| --- | --- | --- |
| Неделя | Описание | Контрольный срок |
| 1 неделя | Получить задание на КП и отсканировать. Оформить шаблон и содержание с указателем страниц, титульный лист. Написать введение. Анализ задания, разработка концепции МПКС, функциональной схемы МПКС. | 3 неделя |
| 2 неделя | Формирование ТЗ.  Написание главы 1  «Формирование технического задания на составную часть эскизного проекта».  Поиск источников информации. Составить библиографические описания для списка использованных источников. | 3 неделя |
| 3 неделя | Согласование и коррекция ТЗ с руководителем.  Изучение теоретических основ. Разработка функциональной схемы криптографического средства. | 5 неделя |
| 4 неделя | Написание главы 2  «Краткие теоретические основы».  Написание главы 3 «Разработка и описание функциональной схемы микропроцессорной  криптографической системы». | 5 неделя |
| 5 неделя | Утверждение ТЗ.  Написание главы 4 «Разработка программного обеспечения микропроцессорного устройства»  (программирование прерываний). | 7 неделя |
| 6 неделя | Написание главы 4  «Разработка программного обеспечения микропроцессорного устройства»  (блоксхемы). | 7 неделя |
| 7 неделя | Алгоритмизация ПО | 9 неделя |
| 8 неделя | Коррекция ТЗ (при необходимости).  Написание главы 4 «Разработка программного обеспечения микропроцессорного устройства»  (блоксхемы). | 9 неделя |
| 9 неделя | Разработка и отладка ПО.  Коррекция блок-схем. Написание главы 4 «Разработка программного обеспечения микропроцессорного устройства»  (текст программ). | 11 неделя  13 неделя |
| 10 неделя | Разработка и отладка ПО.  Коррекция блок-схем. Написание главы 4 «Разработка программного обеспечения микропроцессорного устройства»  (текст программ). | 11 неделя  13 неделя |
| 11 неделя | Разработка и отладка ПО. Написание главы 5  «Анализ результатов работы программного обеспечения микропроцессорного устройства». | 13 неделя |
| 12 неделя | Разработка демонстрационной версии ПО. Написание главы 5  «Анализ результатов работы программного обеспечения микропроцессорного устройства». | 13 неделя |
| 13 неделя | Анализ результатов МПКС.  Выявление недостатков, предложения по модернизации МПКС.  Коррекция ТЗ (при необходимости). | 15 неделя |
| 14 неделя | Составление инструкции по работе с ПО.  Написание заключения.  Оформление приложений. Сдача на проверку. Внесение исправлений (при необходимости). | 15 неделя |
| 15-16 неделя | Подготовка презентации и доклада.  Предварительная защита КП. Сдача на проверку. Внесение исправлений (при необходимости) | 16 неделя |
| 17-18 неделя | Защита КП |  |

|  |  |  |
| --- | --- | --- |
| ПОСЛЕДНИЙ ЛИСТ ТЗ НА СЧ ЭП  студент гр. БЗС2101 МТУСИ  должность, организация разработчика ТЗ  М.С. Рожнов подпись, инициалы, фамилия  «\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_20\_\_\_ г. | | |
| *СОГЛАСОВАНО*  руководитель КП  должность, руководитель проекта  Т.Н. Зуйкова подпись, инициалы, фамилия  «\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_20\_\_\_ г. |  |  |
| *СОГЛАСОВАНО*  студент гр. БЗС2101  должность, исполнитель составной части проекта  Д.Е. Кожевникова подпись, инициалы, фамилия  «\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_20\_\_\_ г. |  |  |
| *СОГЛАСОВАНО*  студент гр. БЗС2101  должность, исполнитель составной части проекта  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ подпись, инициалы, фамилия  «\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_20\_\_\_ г. |  |  |
|  | | |

# ПРИЛОЖЕНИЕ Б

**Текст основной программы шифрования с необходимыми комментариями**

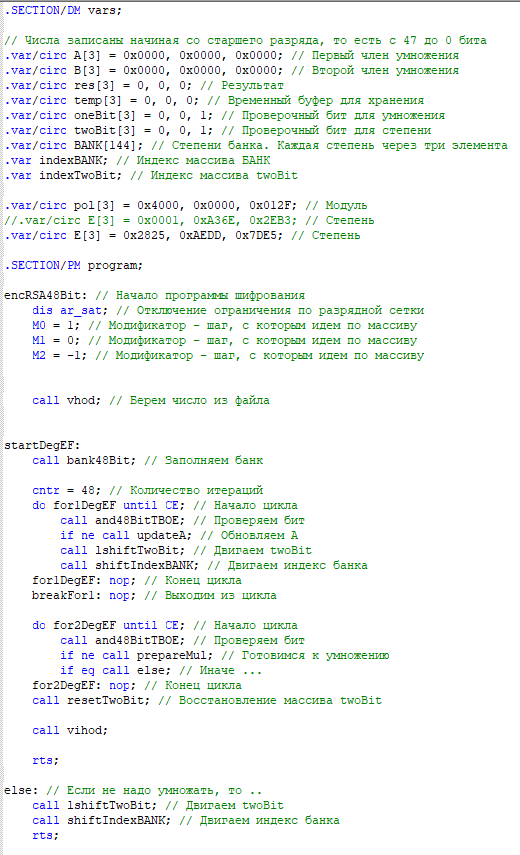
****

Рисунок Б.1 – Текст программы шифрования (начало)

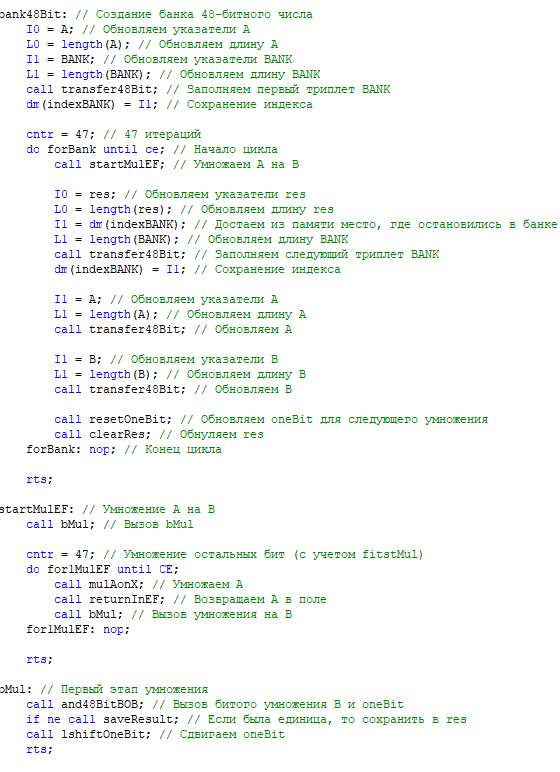


Рисунок Б.2 – Текст программы шифрования (продолжение)

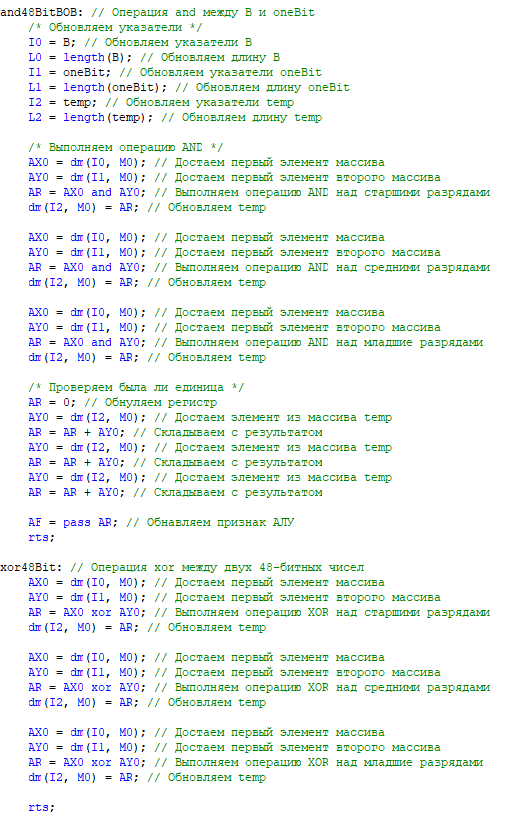


Рисунок Б.3 – Текст программы шифрования (продолжение)

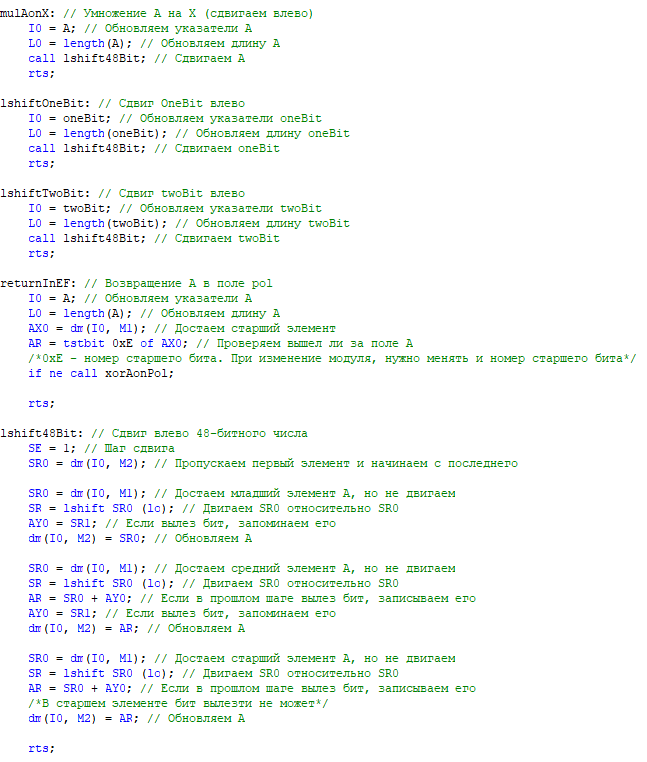


Рисунок Б.4 – Текст программы шифрования (продолжение)

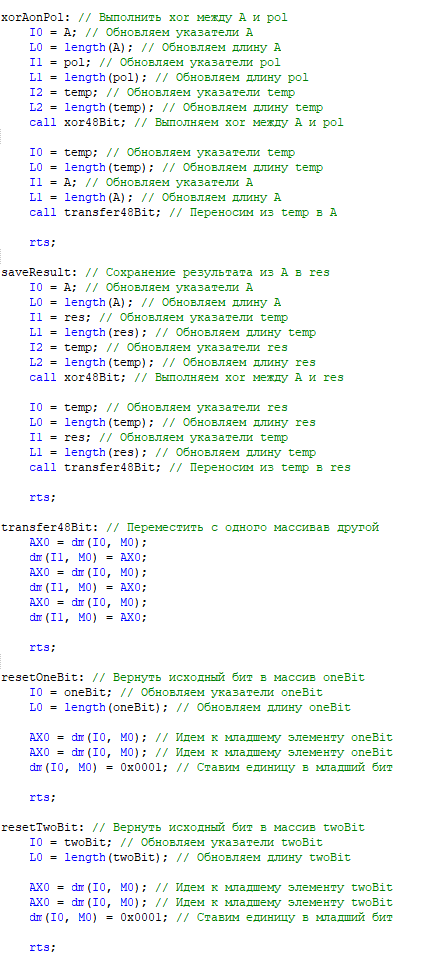


Рисунок Б.5 – Текст программы шифрования (продолжение)

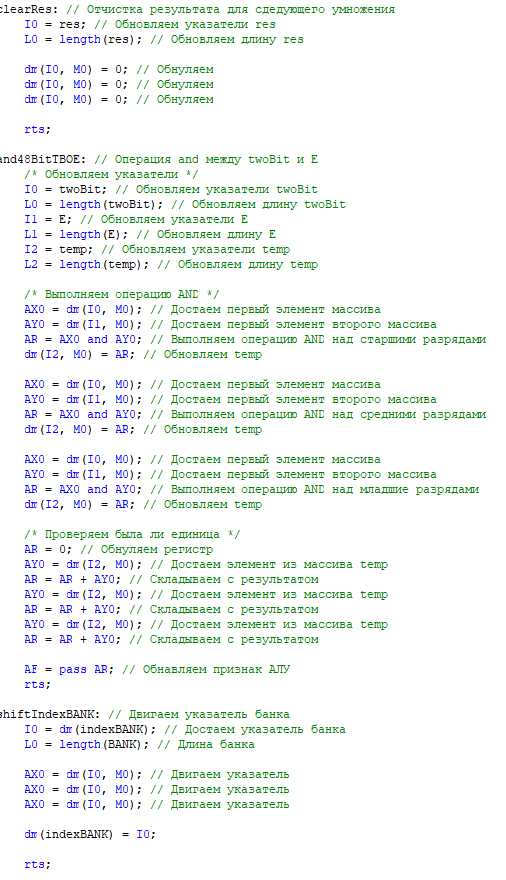


Рисунок Б.6 – Текст программы шифрования (продолжение)

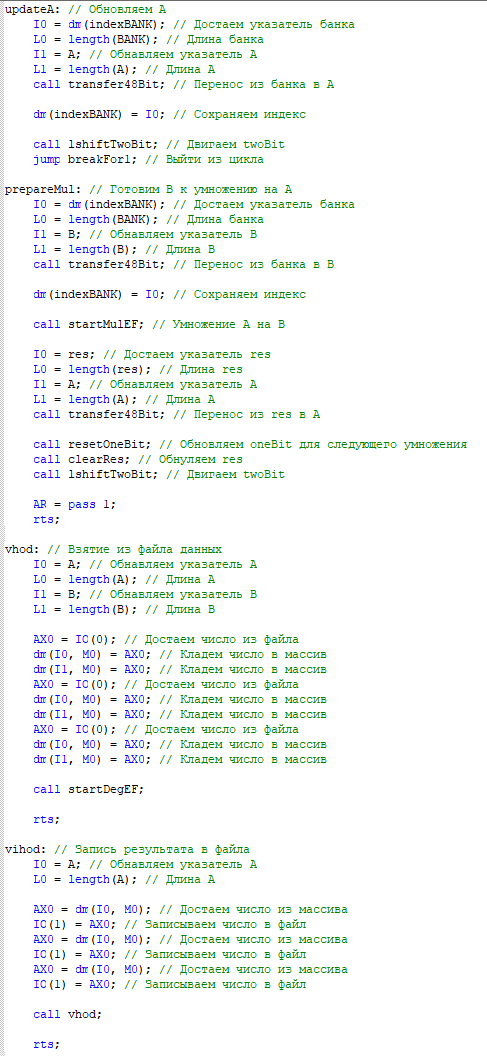


Рисунок Б.7 – Текст программы шифрования (конец)

# ПРИЛОЖЕНИЕ В

**Текст подпрограмм генерации ключей**

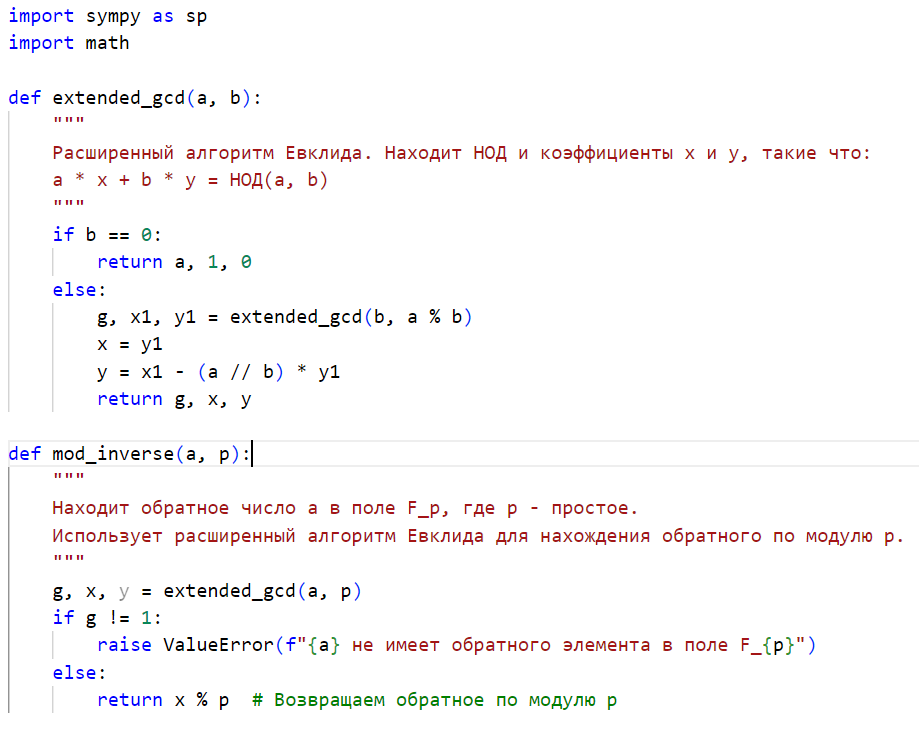
****

Рисунок В.1 – Текст программы генерации ключей (начало)

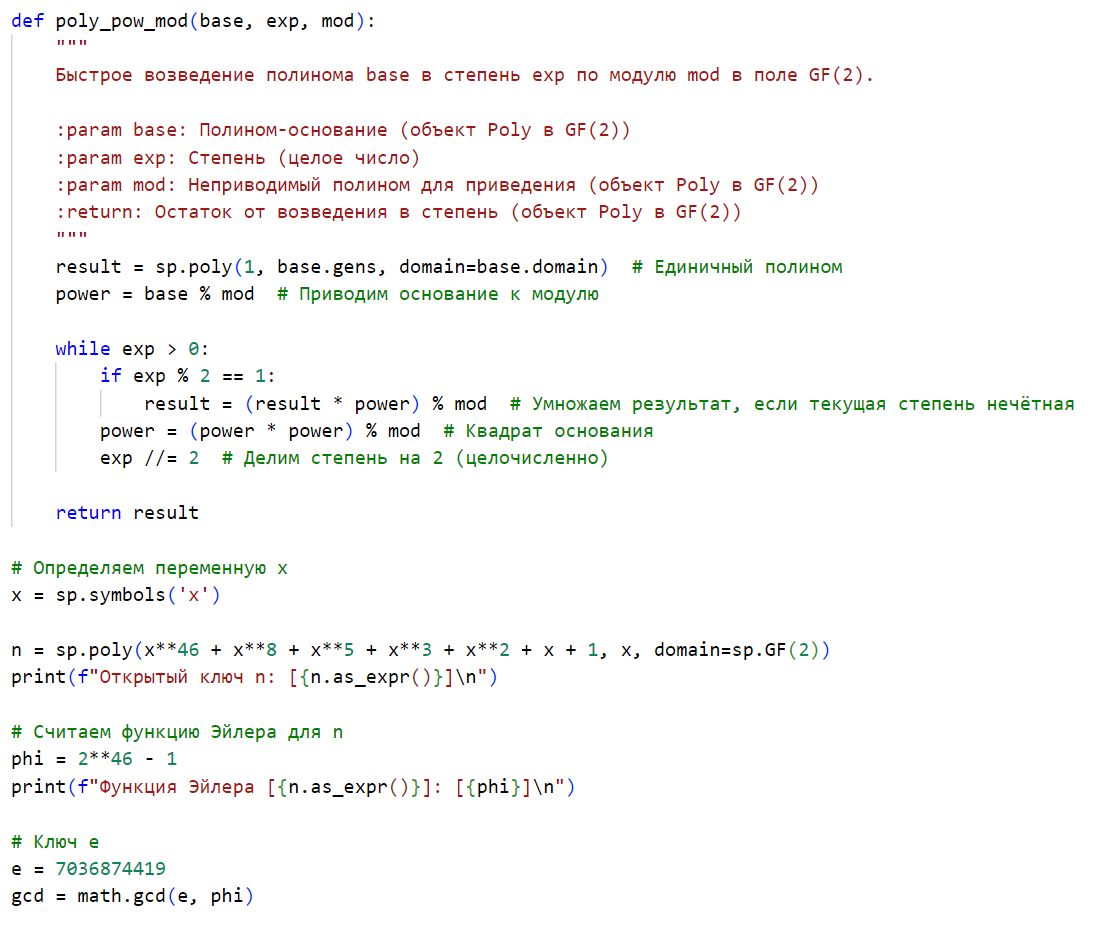
****

Рисунок В.2 – Текст программы генерации ключей (продолжение)

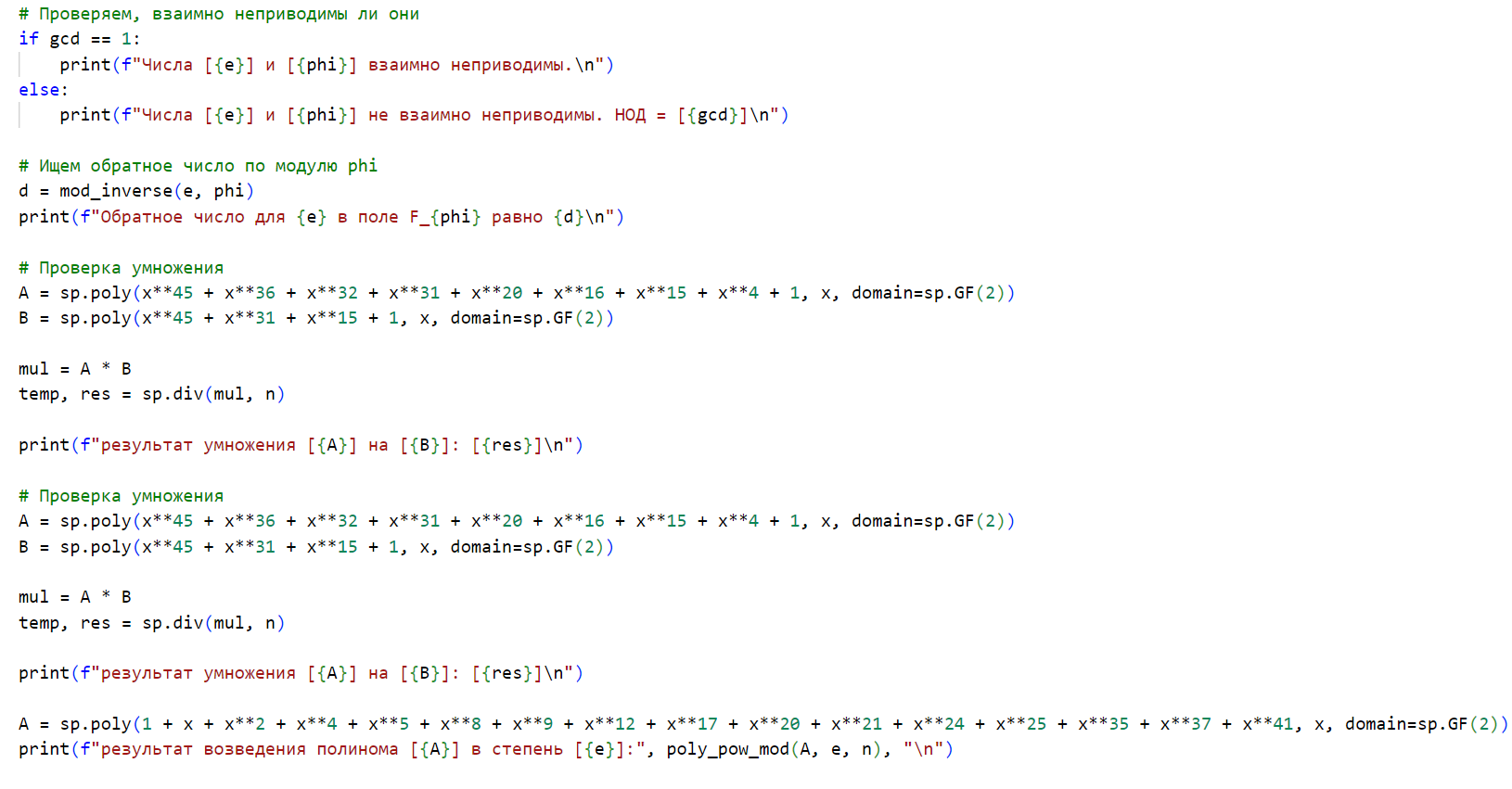


Рисунок В.3 – Текст программы генерации ключей (конец)

# ПРИЛОЖЕНИЕ Г

**Инструкция по работе с МПКС**

Для запуска программы предварительно нужно настроить порты процессора ADSP-2181. Это либо порт SPORT0, либо порты IO.

Следующим шагом следует заполнить массив pol, который является хранилищем для полинома **n.** После этого нужно определить старший бит полинома и редактировать 4 строку подпрограммы returnInEF, а именно номер бита в операции tstbit. Например, если полином является 46-битным, то номер бита в tstbit составляет 0xE.

После следует заполнить массив ключа **e,** который является степенью, в которую будет возводиться сообщение. Заполняется ключ через файл.