

Журнал «Научное обозрение. Технические науки» зарегистрирован Федеральной службой по надзору в сфере связи, информационных технологий и массовых коммуникаций. Свидетельство ПИ № ФС77-57440 ISSN 2500-0799

Двухлетний импакт-фактор РИНЦ – 0,887
Пятилетний импакт-фактор РИНЦ – 0,350

Учредитель, издательство и редакция:
ООО НИЦ «Академия Естествознания»,

Почтовый адрес: 105037, г. Москва, а/я 47
Адрес учредителя: 410056, Саратовская область,
г. Саратов, ул. им. Чапаева В.И., д. 56
Адрес редакции: 410035, Саратовская область,
г. Саратов, ул. Мамонтовой, д. 5

Founder, publisher and edition:
LLC SPC Academy of Natural History,

Post address: 105037, Moscow, p.o. box 47
Founder's address: 410056, Saratov region,
Saratov, 56 Chapaev V.I. str.
Editorial address: 410035, Saratov region,
Saratov, 5 Mamontovoi str.

Подписано в печать 30.06.2023
Дата выхода номера 31.07.2023
Формат 60×90 1/8

Типография
ООО НИЦ «Академия Естествознания»,
410035, Саратовская область,
г. Саратов, ул. Мамонтовой, д. 5

Signed in print 30.06.2023
Release date 31.07.2023
Format 60×90 8.1

Typography
LLC SPC «Academy Of Natural History»
410035, Russia, Saratov region,
Saratov, 5 Mamontovoi str.

Технический редактор Доронкина Е.Н.
Корректор Галенкина Е.С., Дудкина Н.А.

Распространение по свободной цене
Тираж 1000 экз. Заказ НО 2023/3
Подписной индекс в электронном каталоге
«Почта России»: ПА518
© ООО НИЦ «Академия Естествознания»

Журнал «НАУЧНОЕ ОБОЗРЕНИЕ» выходил с 1894 по 1903 год в издательстве П.П. Сойкина. Главным редактором журнала был Михаил Михайлович Филиппов. В журнале публиковались работы Ленина, Плеханова, Циолковского, Менделеева, Бехтерева, Лесгафта и др.

Journal «Scientific Review» published from 1894 to 1903. P.P. Soykin was the publisher. Mikhail Filippov was the Editor in Chief. The journal published works of Lenin, Plekhanov, Tsiolkovsky, Mendeleev, Bekhterev, Lesgaft etc.



М.М. Филиппов (M.M. Philiprov)

С 2014 года издание журнала возобновлено
Академией Естествознания
From 2014 edition of the journal resumed
by Academy of Natural History

Главный редактор: М.Ю. Ледванов
Editor in Chief: M.Yu. Ledvanov

Редакционная коллегия (Editorial Board)
А.Н. Курзанов (A.N. Kurzanov)
Н.Ю. Стукова (N.Yu. Stukova)
М.Н. Бизенкова (M.N. Bizenkova)
Н.Е. Старчикова (N.E. Starchikova)
Т.В. Шнуровозова (T.V. Shnurovozova)

НАУЧНОЕ ОБОЗРЕНИЕ • ТЕХНИЧЕСКИЕ НАУКИ

SCIENTIFIC REVIEW • TECHNICAL SCIENCES

www.science-education.ru

2023 г.



***В журнале представлены научные обзоры,
статьи проблемного
и научно-практического характера***

***The issue contains scientific reviews,
problem and practical scientific articles***

СОДЕРЖАНИЕ

Технические науки (2.2. Электроника, фотоника, приборостроение и связь, 2.3. Информационные технологии и телекоммуникации, 2.4. Энергетика и электротехника)

СТАТЬИ

ОБОСНОВАНИЕ НАДЕЖНОСТИ РАБОТЫ СЕРВОДВИГАТЕЛЕЙ ДЛЯ МАЛОГАБАРИТНЫХ БЕСПИЛОТНЫХ ЛЕТАТЕЛЬНЫХ АППАРАТОВ <i>Дьяков Д.Е., Лиховидов Д.В., Великанов А.В.</i>	5
ПРОГРАММНАЯ СИСТЕМА ИДЕНТИФИКАЦИИ АНОМАЛЬНОЙ АКТИВНОСТИ СУБЪЕКТОВ ИНФОРМАЦИОННО-ТЕЛЕКОММУНИКАЦИОННОЙ СЕТИ <i>Родионов В.Д., Соловьев Н.А.</i>	10
ОБЗОР СУЩЕСТВУЮЩИХ ПОДХОДОВ К МОДЕЛИРОВАНИЮ ПОЛЬЗОВАТЕЛЬСКОГО ИНТЕРФЕЙСА ANDROID <i>Сарычева Ю.Ю., Белов Ю.С.</i>	16
МИКРОПРОГРАММНАЯ РЕАЛИЗАЦИЯ И ЭКСПЕРИМЕНТАЛЬНОЕ ИССЛЕДОВАНИЕ УЧЕБНОГО ПРОЦЕССОРА В MICROSOFT EXCEL <i>Страбыкин Д.А.</i>	21
АЛГОРИТМ КОНТРОЛЯ ВЫПОЛНЕНИЯ ЛОГИЧЕСКИХ ОПЕРАЦИЙ В ПРОГРАММИРУЕМЫХ ЛОГИЧЕСКИХ ИНТЕГРАЛЬНЫХ СХЕМАХ, ВЫПОЛНЯЮЩИХ КРИПТОГРАФИЧЕСКОЕ ПРЕОБРАЗОВАНИЕ ИНФОРМАЦИИ <i>Лукин М.В.</i>	30
ОПТИМИЗАЦИЯ КОНСТРУКЦИИ ФИЛЬТРА СИНТЕЗАТОРА ЧАСТОТ РАДИОКОМПЛЕКСА СВЯЗИ КОСМИЧЕСКОГО АППАРАТА <i>Шмелькова А.А., Андреев В.В., Гурин В.М., Драч В.Е.</i>	40

CONTENTS

Technical sciences (2.2. Electronics, photonics, instrumentation and communication, 2.3. Information technologies and telecommunications, 2.4. Energy and electrical engineering)

ARTICLES

SUBSTANTIATION OF RELIABILITY OF SERVOMOTORS FOR SMALL-SIZED UNMANNED AERIAL VEHICLES <i>Dyakov D.E., Likhovidov D.V., Velikanov A.V.</i>	5
SOFTWARE SYSTEM FOR IDENTIFICATION OF ABNORMAL ACTIVITY OF SUBJECTS OF THE INFORMATION AND TELECOMMUNICATION NETWORK <i>Rodionov V.D., Solovyov N.A.</i>	10
REVIEW OF EXISTING APPROACHES TO MODELING THE ANDROID USER INTERFACE <i>Sarycheva Yu.Yu., Belov Yu.S.</i>	16
MICROPROGRAM IMPLEMENTATION AND EXPERIMENTAL TEST OF THE TRAINING MICROPROCESSOR IN MICROSOFT EXCEL <i>Strabykin D.A.</i>	21
ALGORITHM FOR CONTROLLING THE EXECUTION OF LOGICAL OPERATIONS IN PROGRAMMABLE LOGIC INTEGRATED CIRCUITS PERFORMING CRYPTOGRAPHIC TRANSFORMATION OF INFORMATION <i>Lukin M.V.</i>	30
FILTER DESIGN OPTIMIZATION OF THE FREQUENCY SYNTHESIZER OF THE SPACECRAFT COMMUNICATION RADIO COMPLEX <i>Shmelkova A.A., Andreev V.V., Gurin V.M., Drach V.E.</i>	40

СТАТЬИ

УДК 623.746.4-519

**ОБОСНОВАНИЕ НАДЕЖНОСТИ РАБОТЫ СЕРВОДВИГАТЕЛЕЙ
ДЛЯ МАЛОГАБАРИТНЫХ БЕСПИЛОТНЫХ
ЛЕТАТЕЛЬНЫХ АППАРАТОВ**

Дьяков Д.Е., Лиховидов Д.В., Великанов А.В.

*¹Военный учебно-научный центр Военно-воздушных сил «Военно-воздушная академия имени профессора Н.Е. Жуковского и Ю.А. Гагарина», Воронеж,
e-mail: snooker646@rambler.ru*

Одной из ключевых задач при проектировании малогабаритных беспилотных летательных аппаратов (МБЛА) является обеспечение приемлемых и сертифицируемых показателей надежности для принятых компонентов коммерческих готовых систем (КГС). Шарикоподшипники и втулочные подшипники являются наиболее распространенным типом подшипников, используемых в двигателях. Изношенные подшипники, вибрация, перегрузки по току и загрязненные потенциометры являются причинами возникновения отказов, в дополнение к износу компонентов. В статье представлены экспериментальные результаты, полученные для количественной оценки надежности сервомоторов МБЛА путем регистрации их времени до отказа в определенном наборе тестовых запусков. Сервоприводы нашли широкое применение в современных высокоточных системах управления подвижными объектами. Модель прогнозирования Вейбулла используется для количественного анализа и связанных с ним ключевых математических моделей и может быть применена к широкому спектру данных. Нароботка до отказа, количество циклов до отказа, пробег до отказа, механическое повреждение или аналогичные непрерывные параметры должны быть зафиксированы для всех объектов. Описана методология, принятая для проведения испытаний, использованная в эксперименте. Подробно рассмотрены элементы конструкции, их взаимосвязь и порядок работы на беспилотном летательном аппарате.

Ключевые слова: беспилотный летательный аппарат, сервопривод, модель Вейбулла, вероятность, надежность, испытания

**SUBSTANTIATION OF RELIABILITY OF SERVOMOTORS
FOR SMALL-SIZED UNMANNED AERIAL VEHICLES**

Dyakov D.E., Likhovidov D.V., Velikanov A.V.

*Military Educational and Scientific Centre of the Air Force N. E. Zhukovsky and Y. A. Gagarin
Air Force Academy, Voronezh, e-mail: snooker646@rambler.ru*

One of the key tasks in the design of small-sized unmanned aerial vehicles (UAVs) is to ensure acceptable and certifiable reliability indicators for accepted components of commercial off-the-shelf systems (CGS). Ball bearings and sleeve bearings are the most common type of bearings used in engines. Worn bearings, vibration, overcurrent and dirty potentiometers are among the causes of failures, in addition to component wear. The article presents experimental results obtained to quantify the reliability of MBLA servomotors by recording their time to failure in a certain set of test runs. Servos are widely used in modern high-precision control systems for moving objects. The Weibull prediction model is used for quantitative analysis and related key mathematical models and can be applied to a wide range of data. Operating time to failure, number of cycles to failure, mileage to failure, mechanical damage or similar continuous parameters must be recorded for all objects. The methodology adopted for conducting tests and used for the experiment is described. The design elements, their interrelation and the order of operation on an unmanned aerial vehicle are considered in detail.

Keywords: unmanned aerial vehicle, servo drive, weibull model, probability, reliability, testing

Беспилотные летательные аппараты широко используются в гражданской и военной сферах. Для того чтобы интегрировать их в гражданское воздушное пространство, необходимо обеспечить надежность используемых компонентов.

Малогабаритные быстродействующие сервоприводы применяются в современных высокоточных системах управления подвижными объектами: рулевыми системами летательных аппаратов, автоматическими манипуляторами, роботами с подвижными элементами конструкции и др. [1]. Для правильного выполнения процедуры синтеза системы управления подобных элементов

необходима информация о значениях параметров и структуре математического обоснования надежности работы сервопривода.

Надежность является ключевым фактором в процессе сертификации для оценки безопасности МБЛА. В настоящее время количество отказов МБЛА на один полет выше, чем у обычных пилотируемых самолетов. Феноменальное количество аварий беспилотных летательных аппаратов вызвано несовершенством систем управления полетом, двигательной установки и ошибками оператора [2]. Оценка общей надежности МБЛА имеет решающее значение, поскольку во многих продуктах КГС отсутствует

необходимая информация о надежности. Основное внимание в этом исследовании уделяется оценке надежности компонентов, используемых в МБЛА. МБЛА обычно весят менее 2,4 кг, могут запускаться вручную при размахе крыльев менее 150 мм.

Постановка задачи

Надежность серводвигателей определяется как вероятность того, что изделие будет выполнять свою предназначенную функцию в течение определенного интервала времени при заданных условиях [3]. Надежность может быть определена количественно несколькими способами, включая определение вероятности отказа, частоты отказов и среднего времени наработки на отказ (T_o). Нарботка на отказ определяется как среднее время, в течение которого система будет работать до тех пор, пока не произойдет сбой, или как величина, обратная частоте отказов.

T_o задается с помощью:

$$T_o = \text{оперативное время / отклонение. (1)}$$

На общую надежность системы влияет надежность ее подсистем. При испытании на отказ все компоненты тестируются до тех пор, пока они не выйдут из строя [4, 5]. Обратной связью о принятом положении рычага серводвигателя является широтно-модулированный импульсный сигнал. Бесщеточные двигатели постоянного тока, используемые в серводвигателях, обладают более высокой надежностью, чем их щеточные аналоги, поскольку щетки подвержены эрозии на коллекторе. Серводвигатели, тестируемые в рамках этого исследования, используют для управления полетом системы МБЛА.

Модель Вейбулла является одним из наиболее часто используемых распределений вероятности в области теории надежности. Распределение Вейбулла довольно универсально, охватывает путем варьирования параметров широкий диапазон случаев изменения вероятностей. Наряду с логарифмическим нормальным распределением оно удовлетворительно описывает наработку деталей по усталостным разрушениям, наработку до отказа подшипников, радиодеталей. Используется для оценки надежности деталей и узлов машин, в частности автомобилей, подъемно-транспортных и других машин. Применяется также для оценки надежности по приработочным отказам [6]. Основное преимущество анализа Вейбулла заключается в том, что он обеспечивает точную оценку отказов даже при относительно небольших выборках. Функция надежности задается формулой:

$$P_{(t)} = e^{-(t/\eta)^\beta} \quad (2)$$

Функция плотности распределения ($P_{(c)}$) и функция плотности вероятности ($P_{(p)}$) выражаются как:

$$P_{(c)} = 1 - e^{-(t/\eta)^\beta} \quad (3)$$

$$P_{(p)} = \frac{\beta}{\eta} \left(\frac{t}{\eta} \right)^{\beta-1} e^{-(t/\eta)^\beta}, \quad (4)$$

где β – параметр формы Вейбулла, η – параметр шкалы Вейбулла, соответствующий времени, когда 63% выборки выходят из строя, а t – временные рамки. Распределения Вейбулла могут гибко представлять широкий спектр других распределений, включая экспоненциальное, нормальное, рэлеевское, пуассоновское и биномиальное. Однако одна кривая Вейбулла не способна отразить все три стадии, а именно увеличение, уменьшение и постоянство отказов [7]. При прогнозировании надежности электронного оборудования применяют два параметра Вейбулла для моделирования надежности оборудования, а прогнозируемая частота отказов (отказы/106 ч) выражается как:

$$\lambda_{(t)} = \left(\frac{t^2}{\alpha_B^3} + \frac{1}{\alpha_W} \right) \times 10^6, \quad (5)$$

где α_B^3 – срок службы подшипников двигателя по характеристике Вейбулла и α_W – характеристика Вейбулла по сроку службы обмоток двигателя. Предполагая стандартную частоту отказов $2,3 \times 10^{-5}$ на час полета, равную наработку на отказ 43 478 часов, мы получаем количество отказов/106 часов. Неисправность определяется как состояние, при котором серводвигатель больше не способен поднимать грузы с заданной частотой. Режим неисправности определяется с помощью визуального осмотра после того, как возникла неисправность. Неисправности подшипников приводят к существенной потере работоспособности серводвигателя.

Тестирование успешного запуска выполняется, когда ожидается, что все компоненты выдержат тестирование с нулевыми сбоями. Тестирование успешного запуска используется для проверки определенного критерия надежности, когда ожидается, что в течение ограниченного по времени тестирования не произойдет никаких сбоев. Определяется количество выборок, необходимых для успешного выполнения теста, и на основе биномиального распределения, где p представляет вероятность неудачи, ве-

роятность получения k плохих элементов и $(n - k)$ хороших элементов определяется по формуле [7]:

$$f(k) = \frac{n!}{k!(n-k)!} p^k (1-p)^{n-k}. \quad (6)$$

Были приняты разумные значения надежности 80% при доверительном уровне C 50%, что привело к размеру выборки в семь человек с фактическим уровнем достоверности 52% и уровнем достоверности, выраженным как:

$$R = 1 - p \quad (7)$$

$$C = 1 - \sum_{i=0}^k \frac{N!}{i!(N-i)!} R^{N-i} (1-R)^i. \quad (8)$$

Все компоненты запускались непрерывно до отказа, и время до отказа регистрировалось. Моделирование в MATLABM допускало изменение входных данных для серводвигателя, а именно частоты цикла и частоты обновления. Количество запусков зависит от требований к питанию. В условиях, когда для компонента требовалось несколько запусков, порядок выбирается случайным образом, чтобы уменьшить влияние любых неизвестных переменных. Также регистрировалась температура окружающей среды, поскольку она влияет на срок службы изоляции и увеличение сопротивления проводников. Была выбрана самая высокая частота обновления, чтобы обеспечить серводвигатели с более плавным непрерывным движением во время цикла. При работе серводвигателей на максимальном напряжении серводвигатели работали под более высоким напряжением. Это сократило время, необходимое для проведения эксперимента, поскольку время до отказа зависит от стресса. Частота цикла серводвигателей выбрана как минимально возможная, чтобы серводвигатели могли достигать всего диапазона значений угла даже при нагрузке. Условия нагрузки были выбраны таким образом, чтобы представлять усилия, которые можно было ожидать от МБЛА. Испытательный крутящий момент τ задается формулой:

$$\tau = \frac{\left(\frac{\sin(S) \tan(S)}{\tan(n)} \right) M_p C^2 L V^2}{4RT}, \quad (9)$$

где $S=30^\circ$ – угол наклона поверхности управления от нейтралы, $n=45^\circ$ – угол отклонения рычага серводвигателя от нейтралы, $M=28,6$ г/моль – молекулярная масса воздуха, p – давление воздуха, $C=60$ мм – хорда, $L=440$ мм – длина поверхности управления,

$V=40$ м/с – скорость полета, T – температура в градусах Цельсия и является постоянной идеального газа и равна $8,31451$ ДжК⁻¹моль⁻¹. С учетом характеристик МБЛА, такого как JAVELIN [7], крутящий момент (τ) гр.см получается при максимальной скорости 40 м/с. Полученные значения крутящего момента элеронов и отклонения поверхности сравниваются. Принятые допущения и ограничения: условия окружающей среды, включая уровень моря, нулевую влажность и температуру 12°C , нулевые ускорения и угловые скорости, угол атаки крыла, стабилизатора, предохранителя и поверхностей управления – равны нулю, рычаги управления имеют нулевое смещение по линии шарнира, механизмы управления без трения, поверхности сбалансированы по массе, аэродинамические противовесы нулевые.

Экспериментальная часть

Для испытания использовались семь цифровых серводвигателей НК15148В [8]. Дополнительно применялись GW Instek SPS3610, Manson NP-9615, 5 м финишной проволоки весом 5,2 кг, семь грузиков по 500 г каждый, микроконтроллер, портативный компьютер, веб-камера, три соединительных провода макетной платы и три зажима типа «крокодил». Для наблюдения за тестированием использовалась веб-камера с разрешением 640x480 пикселей, которой управлял скрипт на python. Частота кадров в секунду (FPS) выбирается на основе параметров периода работы серводвигателя (1 сек) и размера используемого видеофайла.

Зафиксированы периоды отказов и соответствующее время до отказа. Результирующий логарифмический график соответствия модели Вейбулла показан на рисунке 1 [7].

Точное время выхода из строя 2-го, 5-го и 11-го серводвигателей зафиксировано между 3,5 и 4,5 часами из-за чрезмерного изменения условий освещения, которые повлияли на точный сбор данных, и представлено в параметр формы Вейбулла $\beta=0,7986$, параметр шкалы Вейбулла $\eta=25,71$ получены в пределах 95%-ного доверительного интервала. По результатам испытаний под нагрузкой [9] получена наработка на отказ серводвигателя, которая составила 29 часов и 12 минут.

Анализ вышедших из строя серводвигателей позволяет получить представление о причинах отказов. Значение параметра шкалы функции Вейбулла ниже единицы указывает на то, что частота отказов уменьшается с течением времени и что в серводвигателях наблюдается значительная частота отказов: пять из семи выходят из строя менее чем за 24 часа.

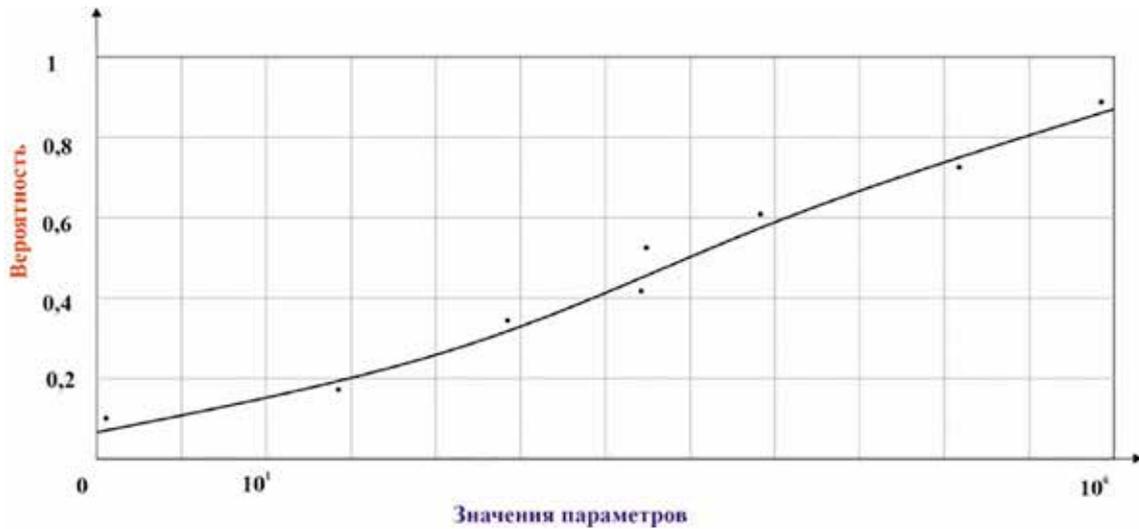


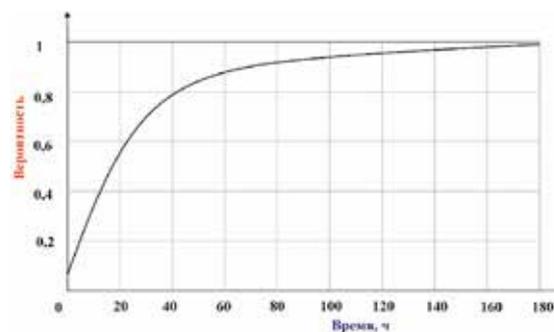
Рис. 1. Результаты испытаний серводвигателей по модели Вейбулла

Результирующая наработка на отказ серводвигателей не соответствует требованиям, предъявляемым к пилотируемым самолетам, и значительно ниже оценки в 43 478 часов.

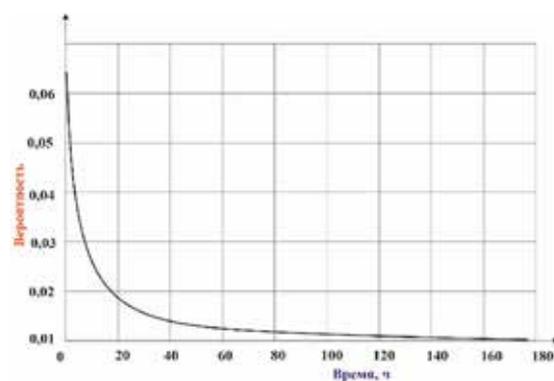
Надежность является одним из ряда факторов, влияющих на риск, который беспилотные летательные аппараты представляют для людей и имущества на земле. Если предположить, что отказ сервопривода является «катастрофическим» (т.е. приводит к потере управления воздушным судном), то результаты по надежности указывают на уровень надежности, ожидаемый для недорогих сервоприводов. Такие сервоприводы могли бы использоваться для операций беспилотных летательных аппаратов с низким уровнем риска (например, небольших беспилотных летательных аппаратов, работающих в малонаселенных регионах) и там, где экономика бизнес-модели допускает более высокие показатели потерь платформы.

На рисунках 2а и 2б показаны результаты функции плотности распределения ($P_{(p)}$) и функции плотности вероятности ($P_{(c)}$) для протестированных серводвигателей соответственно.

Для данных испытаний разрабатываются методология и испытательный стенд. Хотя параметры недорогих серводвигателей не отвечают применимым стандартам сертификации, делается вывод, что параметры надежности могут быть достаточными для выполнения требования к операциям беспилотных летательных аппаратов с низким уровнем риска (например, полеты небольших беспилотных летательных аппаратов над малонаселенными районами).



а)



б)

а) функция плотности распределения $P_{(c)}$
 б) функция плотности вероятности $P_{(p)}$

Рис. 2. Результаты испытаний серводвигателей

Будущие исследования будут посвящены внедрению компонентов в различные авиационные системы и системы управления полетом, включая распознавание и уклонение, выработку (распределение)

и хранение электроэнергии, лазерное предотвращение препятствий и мониторинг, интегрированную навигацию и наведение [10], и системы повышения целостности.

Заключение

Таким образом, были представлены экспериментальные мероприятия, проведенные для оценки надежности недорогих серводвигателей КГС. Полученные результаты указывают на уровень надежности, ожидаемый для недорогих сервоприводов. Такие сервоприводы можно было бы использовать для операций с МБЛА с низким уровнем риска (например, для работающих в малонаселенных регионах) и там, где экономика бизнес-модели допускает более высокие показатели потерь. Для серводвигателей было зафиксировано время до отказа, подверженное серьезными нагрузками во всем диапазоне операций. Для получения профиля частоты отказов была принята модель Вейбулла. Расчетная наработка на отказ для подобных серводвигателей составила 29 часов и 12 минут.

Список литературы

1. Котов К.Ю., Мальцев А.С., Нестеров А.А., Соболев М.А., Филиппов М.Н., Ян А.П. Определение параметров сервопривода на основе экспериментальных данных // Проблемы управления и моделирования в сложных системах: труды XVI Международной конф. Самара: Самарский научный центр РАН, 2014. С. 592-596.
2. Dermentzoudis M., Establishment of Models and Data Tracking for Small UAV Reliability, Master's thesis, Naval Postgraduate School, Monterey, California, USA, 2004. 225 p.
3. Гнеденко Б.В., Беляев Ю.К., Соловьев А.Д. Математические методы в теории надежности. Москва: Наука, 1965. 524 с.
4. Котов К.Ю., Мальцев А.С., Нестеров А.А., Соболев М.А., Филиппов М.Н., Ян А.П. Разработка и идентификация модели сервопривода на основе экспериментальных данных // Автометрия. 2013. Т. 49, № 6. С. 42-50.
5. Котов К.Ю., Мальцев А.С., Нестеров А.А., Соболев М.А., Филиппов М.Н., Ян А.П. Стенд для разработки и идентификации моделей сервоприводов // Индустриальные информационные системы – ИИС-2013. Всероссийская конференция. Новосибирск: КТИ ВТ СО РАН, 2013. С. 33-34.
6. Murthy D.N.P., Xie M., Jiang R., Weibull Models, Wiley, New York, USA, 2003. 383 p.
7. Burston M., Sabatini R., Gardi A., Clothier R. Reverse engineering of a fixed wing Unmanned Aircraft 6-DoF model based on laser scanner measurements. 2014. DOI: 10.1109/MetroAeroSpace.2014.6865910.
8. Цифровой серводвигатель НК15148В. [Электронный ресурс]. URL: https://hobbyking.com/ru_ru/hobbykingtm-hk15148b-digital-servo-2-8kg-0-14sec-19g.html. (дата обращения: 09.04.2023).
9. Дьяков Д.Е., Валиулин И.А., Лиховидов Д.В., Великанов А.В. Создание действующей модели безводильного аэродромного буксировщика // Вестник Брянского государственного технического университета. 2021. № 10(107). С. 45-52. DOI: 10.30987/1999-8775-2021-10-45-52.
10. Burston M., Sabatini R., Gardi A., Clothier R. Avionics sensor fusion for small size unmanned aircraft Sense-and-Avoid. 2014. DOI: 10.1109/MetroAeroSpace.2014.6865933.

УДК 004.02

ПРОГРАММНАЯ СИСТЕМА ИДЕНТИФИКАЦИИ АНОМАЛЬНОЙ АКТИВНОСТИ СУБЪЕКТОВ ИНФОРМАЦИОННО-ТЕЛЕКОММУНИКАЦИОННОЙ СЕТИ

Родионов В.Д., Соловьев Н.А.

*ФГБОУ ВО «Оренбургский государственный университет», Оренбург,
e-mail: vvvddrrr@gmail.com*

Статья посвящена разработке программной системы, предназначенной для идентификации аномальной активности субъектов информационно-телекоммуникационной сети. Для достижения этой цели проведен системный анализ, который позволил выявить необходимость автоматизации информационных процессов, а также актуальность создания новой системы защиты, основанной на анализе аналогов существующих средств информационной безопасности. Основой для этой системы послужил спектральный анализ сетевого трафика, который позволяет обнаружить аномалии и сетевые атаки. Для обнаружения аномалий в данной системе используется расчет коэффициентов корреляции Пирсона, которые показывают степень схожести частотных показателей сетевого трафика на разных временных интервалах. Для расчета этих коэффициентов используется дискретное оконное преобразование Фурье. В статье показан алгоритм работы программной системы, который позволяет достичь требуемой эффективности обнаружения аномалий в информационно-телекоммуникационной сети. Для проверки достоверности проводимых расчетов сравниваются коэффициенты корреляции Пирсона в предложенной системе и в программной среде MatLab при аналогичных входных данных. Исследование показало, что разработанная система обеспечивает высокую точность в обнаружении аномалий объектов информационно-телекоммуникационной сети. Программа может определить как аномалию, так и конкретную атаку, сравнив частотные показатели текущего сетевого трафика с записанным шаблоном.

Ключевые слова: обнаружение аномалий, сетевой трафик, Фурье-анализ, коэффициент корреляции Пирсона, информационная безопасность

SOFTWARE SYSTEM FOR IDENTIFICATION OF ABNORMAL ACTIVITY OF SUBJECTS OF THE INFORMATION AND TELECOMMUNICATION NETWORK

Rodionov V.D., Solovyov N.A.

Orenburg state university, Orenbug, e-mail: vvvddrrr@gmail.com

This article is devoted to the development of a software system designed to identify abnormal activity of subjects of the information and telecommunications network. To achieve this goal, a system analysis was carried out, which revealed the need for automation of information processes, as well as the relevance of creating a new security system based on the analysis of analogues of existing information security tools. The basis for this system was the spectral analysis of network traffic, which allows you to detect anomalies and network attacks. To detect anomalies in this system, the calculation of Pearson correlation coefficients is used, which show the degree of similarity of the frequency indicators of network traffic at different time intervals. A discrete windowed Fourier transform is used to calculate these coefficients. The article shows the algorithm of the developed software system, which allows to achieve high efficiency in detecting anomalies in the information and telecommunications network. To verify the reliability of the calculations carried out, the Pearson correlation coefficient was calculated in the MatLab software environment with similar input data. The study showed that the developed system works efficiently and provides high accuracy in detecting anomalies in the information and telecommunications network. The program can determine both an anomaly and a specific attack by comparing the frequency indicators of the current network traffic with the recorded pattern.

Keywords: anomaly detection, network traffic, Fourier analysis, Pearson correlation coefficient, information security

С каждым годом вопрос использования средства обнаружения сетевых атак становится все актуальнее, так как растет количество сфер, в которых применяются сетевые технологии, а также увеличивается количество пользователей, использующих сетевые технологии. Рост сетевых технологий влияет на рост количества методов сетевых атак. Использование средств, обнаруживающих сетевые атаки по заранее прописанным правилам, становится неактуальным, так как новые типы атак не будут идентифицированы.

Компания Positive Technologies в статье «Обнаружение распространенных угроз

ИБ в сетевом трафике» приводит результаты исследования угроз в 60 организациях за 2021–2022 годы. Доля компаний, в которых была обнаружена подозрительная сетевая активность, составляет 93% [1].

Без использования средств обнаружения сетевых атак компании рискуют потерять прибыль и доверие у своих пользователей из-за таких последствий, как отказ в обслуживании и утечка данных.

Оперативное обнаружение защищает от последствий сетевых атак, что доказывает необходимость разработки программной среды, способной обнаруживать аномалии и новые типы сетевых атак.

Чаще всего для защиты информационно-телекоммуникационной сети применяются системы обнаружения сетевых атак Snort и Suricata. Они используют правила-шаблоны, которые контролируют сетевые пакеты, и уведомляют администратора, если сетевой трафик соответствует заранее прописанным правилам. Такой подход исключает выявление новых типов сетевых атак, а также требует жесткой редакции правил, что осложняет администрирование таких средств.

Цель исследования – автоматизация информационных процессов обнаружения сетевых атак на основе анализа сетевого трафика. Для этого определены показатели, которые используются при обнаружении сетевых атак, создана математическая модель процесса идентификации сетевых атак и аномалий и разработано программное средство, реализующее предложенный метод.

Методы обнаружения сетевых атак

При рассмотрении методов обнаружения сетевых атак по способу интерпретации входных данных можно выделить два класса [2]:

- обнаружение аномалий;
- обнаружение злоупотреблений.

Методы обнаружения аномалий подразумевают поиск отклонений от обычного трафика внутри сети. Сложность данного метода заключается в определении допустимого отклонения и нормальной активности.

Методы обнаружения злоупотреблений заключаются в идентификации несанкционированных действий путем сравнения с шаблоном атаки. Шаблон атаки – совокупность действий, соответствующих поведению конкретной атаки. Параметры сетевого трафика отправляются на проверку, в которой происходит сравнение с различными шаблонами атаки и определяется, осуществляется атака или нет. Данный метод позволяет сразу узнать тип атаки, так как для каждой атаки разрабатывается свой шаблон, но это создает и определенные проблемы: нужно описывать каждую атаку, что является трудоемким процессом. Если атака не описана, то никаких предупреждений не будет выдано, так как все тесты будут пройдены.

Обнаружение аномалий может происходить двумя способами:

- проверка на соответствие шаблонам;
- спектральный анализ.

Спектральный анализ – метод обработки сетевого трафика, который позволяет описать частотный состав исследуемого

трафика [3]. Быстрое преобразование Фурье переводит временные характеристики сетевого трафика в частотный вид [4].

Использование спектрального анализа позволяет выявлять скрытые закономерности сетевого трафика.

Для обнаружения аномалий в сетевом трафике фрагмент трафика T длиной N отсчетов сравнивается с предыдущим фрагментом $T-1$ такой же длины.

Для сравнения трафика составляются последовательности, которые могут состоять из:

- количества пакетов, распределенных по типу протоколов;
- количества всех пакетов за одну секунду;
- размера сетевого трафика за секунду;
- размера трафика определенного протокола за секунду;
- количества хостов, обнаруженных в сетевом трафике.

С помощью быстрого дискретного преобразования Фурье числовая одномерная последовательность длиной N преобразуется в частотный вид (1). После этого рассчитывается коэффициент корреляции Пирсона, который позволяет оценить схожесть частотных показателей сетевого трафика (2).

$$y(k) = \sum_{n=0}^{N-1} x(n) e^{-2\pi j \frac{kn}{N}} \quad (1)$$

$$R_{xy} = \frac{\sum (X - \bar{X})(Y - \bar{Y})}{\sqrt{\sum (X - \bar{X})^2 \sum (Y - \bar{Y})^2}} \quad (2)$$

Здесь X – количество сетевых пакетов за секунду фрагмента $T-1$; \bar{X} среднее – арифметическое пакетов в секунду для фрагмента $T-1$; Y – количество сетевых пакетов за секунду фрагмента T ; \bar{Y} – среднее арифметическое пакетов в секунду для фрагмента T .

Пороговые значения коэффициентов корреляции и оптимальный период измерений сетевого трафика выявлены экспериментальным путем [5]:

- оптимальный период измерений сетевого трафика – 10 секунд;
- лучший коэффициент корреляции для обнаружения аномалии – 0,3;
- лучший коэффициент корреляции для обнаружения атаки – 0,7.

Разработка программного средства

Понимание функционала разрабатываемой программы отражается в функциональной схеме. Функции разрабатываемой системы представляются в виде специальных объектов и утверждений.

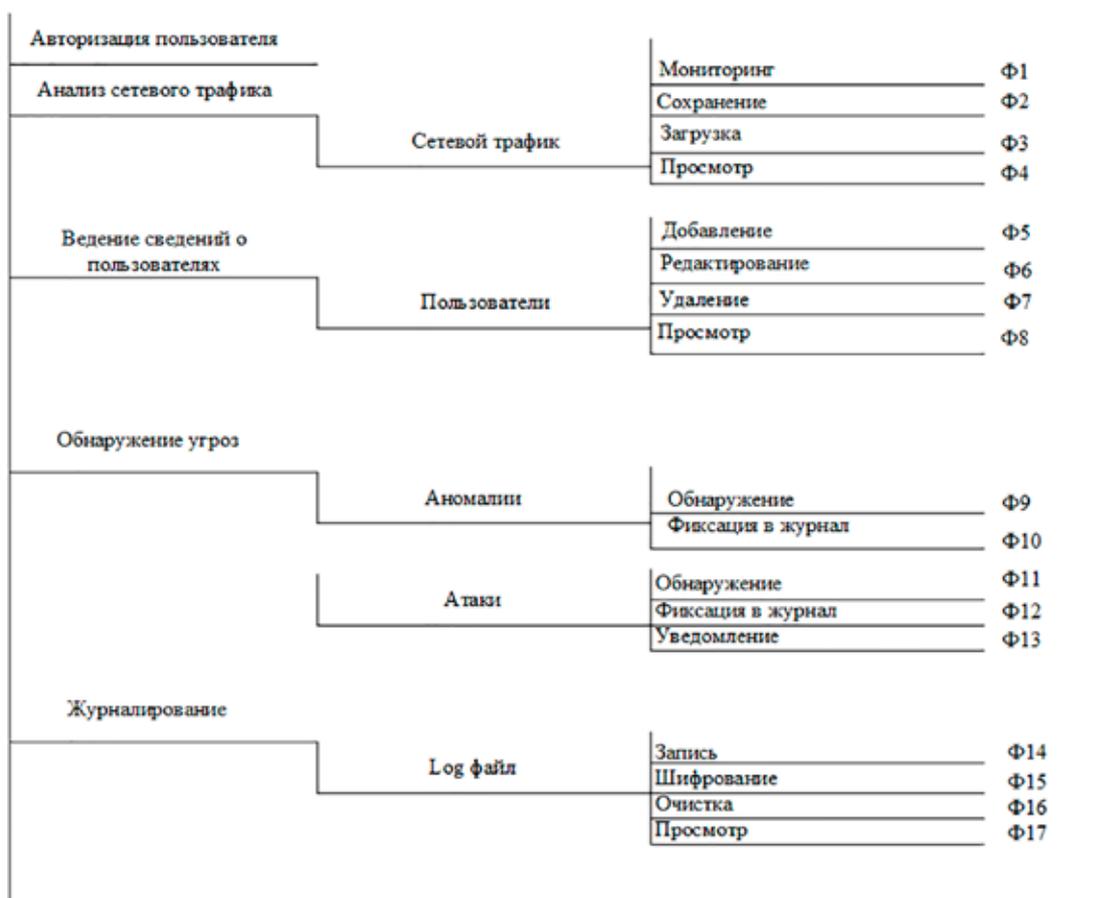


Рис. 1. Иерархия функций разрабатываемой программы

Основными функциями разрабатываемого программного средства являются:

- авторизация пользователей;
- анализ сетевого трафика;
- ведение сведений о пользователях;
- обнаружение угроз;
- журналирование.

Функции, реализуемые разрабатываемой программой, отображены в иерархии функций (рис. 1).

Функции взаимодействуют с зашифрованным журналом событий (.log файлом), с записанным сетевым трафиком (.pcap файл) и базой данных, в которой хранятся сведения о правах и аутентификационные данные пользователей.

Исходя из иерархии функции, построена диаграмма состояний, подробно описывающая процесс работы программы (рис. 2).

На основе выявленных функций и описания состояний была разработана программа на языке программирования Python 3.10 (рис. 3). На главном окне программы показаны четыре графика. Верхний левый график показывает количество пакетов в се-

кунду за прошлый период в 10 секунд. Верхний правый график показывает количество пакетов за текущий период. Преобразованные в частотный вид показатели отображены на нижнем графике. Коэффициент корреляции Пирсона, рассчитанный между частотными показателями двух периодов, показан ниже графиков. Если коэффициент меньше 0,3, то происходит сравнение с частотными показателями, записанными во время атак. При сравнении с шаблонами атак, если коэффициент больше 0,7, появляется окно с оповещением об атаке, в ином случае появляется окно с оповещением об аномалии.

С главного окна пользователю доступна справочная информация о программе: Окна «Помощь», «О программе». Также пользователю доступна загрузка .pcap файла во вкладке «Файл», после которой произойдет анализ выбранного файла.

Во вкладке «Учетные записи» реализовано управление учетными записями системы. Пользователям системы доступны редактирование существующих учетных записей, создание новых и удаление старых.

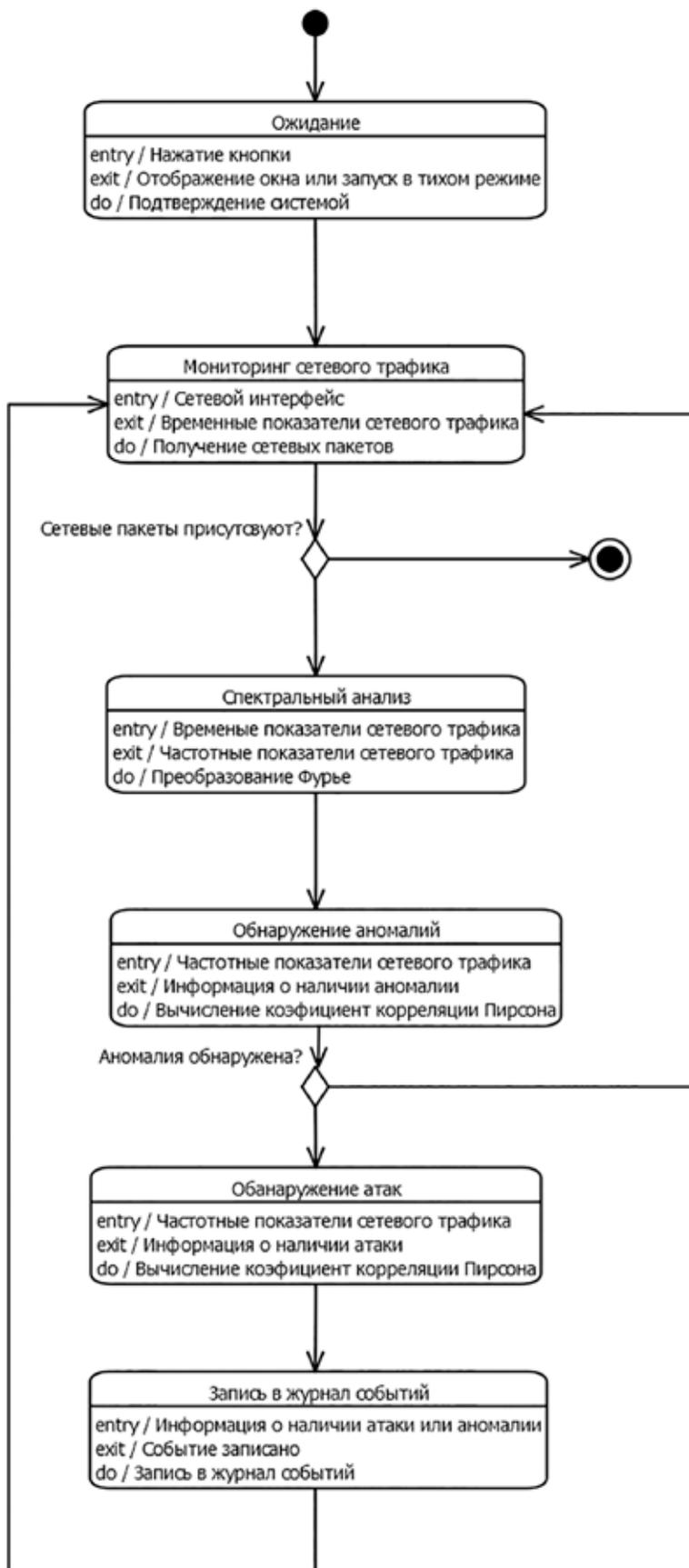


Рис. 2. Диаграмма состояний

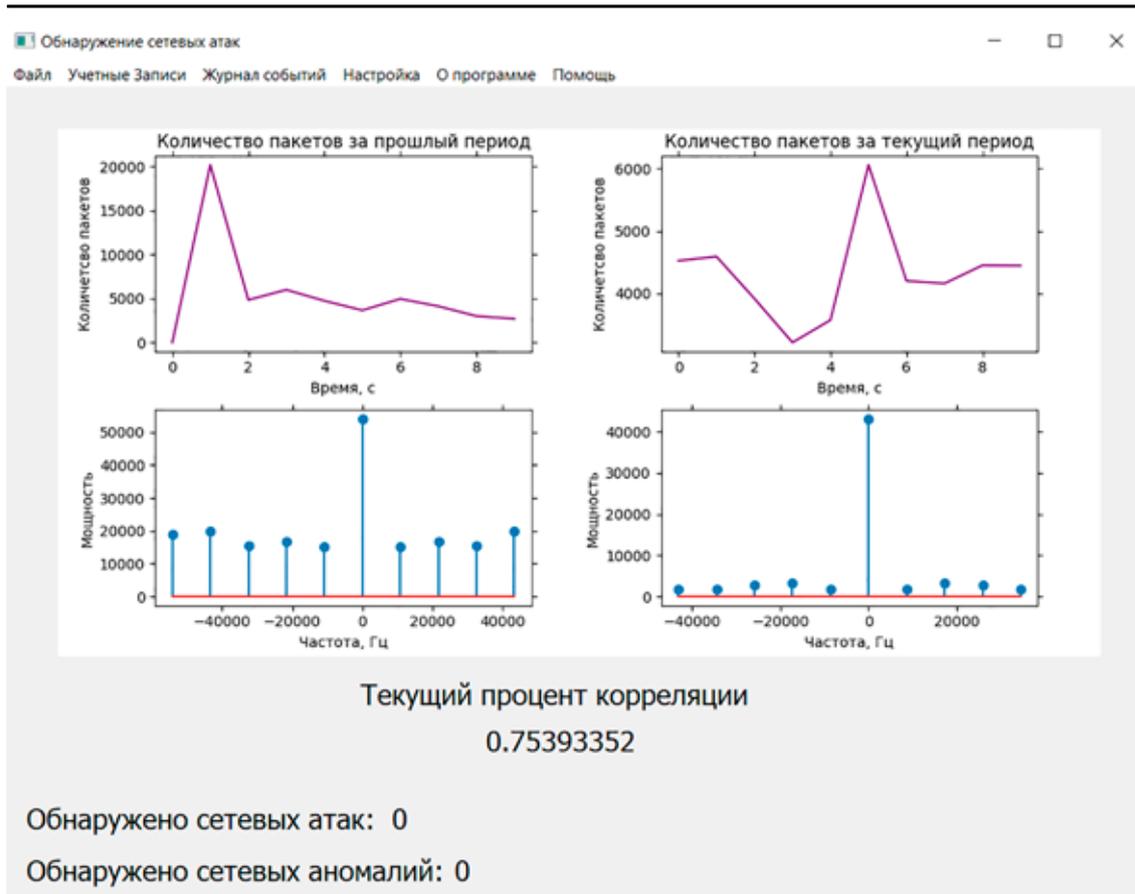


Рис. 3. Интерфейс АИС

```

MAS:= [0 20170 4833 5981 4729 3656 4951 4116 2998 2690]
MAS2:= [4523 4590 3914 3211 3570 6061 4199 4158 4450 4446]
F:=dft(MAS)      D:=dft(MAS2)
F1:=Re(F)       F2:=Re(D)
CORC:=Re(corr(F,D))
CORC=0.754

```

Рис. 4. Результат расчета в MathCAD

Вкладка «Журнал событий» вызывает окно с отображением событий, записанных программой в .log файл. Пользователь может очистить журнал событий, если у него есть соответствующие права на данное действие.

Для проверки результатов, выдаваемых программным средством, произведено сравнение с результатами, рассчитанными в MathCAD. В качестве исходных данных представлены следующие массивы с количеством пакетов за T-1 и T:

```

[0, 20170, 4833, 5981, 4729, 3656, 4951, 4116, 2998, 2690]
[4523, 4590, 3914, 3211, 3570, 6061, 4199, 4158, 4450, 4446]

```

Выходным значением программы является коэффициент корреляции Пирсона (рис. 3). Программа выдала: 0,7539. Встроенная функция в MathCAD dft выполняет преобразование Фурье, а функция $corr$ рассчитывает коэффициент корреляции Пирсона (рис. 4).

MathCAD выдал округленный результат 0,754, что доказывает точность выдаваемого результата разработанного программного средства.

Заключение

Данная статья описывает метод обнаружения сетевых атак и аномалий, основанный на спектральном анализе сетевого трафика. Также описывается разработанное программное средство, которое реализует предложенный метод обнаружения аномалий с помощью спектрального анализа. Основные функции программы включают авторизацию пользователей, ана-

лиз сетевого трафика, обнаружение угроз и журналирование.

Список литературы

1. Обнаружение распространенных угроз ИБ в сетевом трафике. [Электронный ресурс]. URL: <https://www.ptsecurity.com/ru-ru/research/analytics/network-traffic-analysis-2022/> (дата обращения: 15.02.2023).
2. Лукацкий А.В. Обнаружение атак. СПб.: БХВ-Петербург, 2003. 608 с.
3. Браницкий А.А. Обнаружение аномальных сетевых соединений на основе гибридизации методов вычислительного интеллекта: дис. ... канд. тех. наук. Санкт-Петербург, 2018. 305 с.
4. Медведев С.Ю. Преобразование Фурье и классический цифровой спектральный анализ. [Электронный ресурс]. URL: http://www.vibration.ru/preobraz_fur.shtml (дата обращения: 15.02.2023).
5. Родионов В.Д. Экспериментальное исследование защиты информационно-телекоммуникационной системы на основе спектрального анализа сетевого трафика // Научные исследования студентов и учащихся: сборник статей VIII Международной научно-практической конференции, Пенза, 07 февраля 2023 года. Пенза: Наука и Просвещение (ИП Гуляев Г.Ю.), 2023. С. 75-78.

УДК 004.514

ОБЗОР СУЩЕСТВУЮЩИХ ПОДХОДОВ К МОДЕЛИРОВАНИЮ ПОЛЬЗОВАТЕЛЬСКОГО ИНТЕРФЕЙСА ANDROID

Сарычева Ю.Ю., Белов Ю.С.

ФГБОУ ВО «Московский государственный технический университет имени Н.Э. Баумана»,
Калужский филиал, Калуга, e-mail: yulia.sarycheva99@mail.ru

Графический интерфейс пользователя является неотъемлемой частью программного обеспечения на большинстве основных платформ, включая ОС Android, и представляет широкий интерес для многих исследователей. Конечные пользователи взаимодействуют с приложением в зависимости от того, что они хотят делать с приложением и что они видят в его графическом интерфейсе. Поскольку разные приложения часто используют общие шаблоны проектирования пользовательского интерфейса, интуитивно понятно, что способ взаимодействия людей с графическим интерфейсом можно обобщить для разных приложений. Цель модели взаимодействия состоит в том, чтобы зафиксировать такие обобщенные модели взаимодействия. Приложения для Android состоят из четырех основных типов компонентов: действий, сервисов, приемников вещания и поставщиков контента. В данном исследовании будут рассмотрены основные сущности графического интерфейса пользователя в приложениях на базе ОС Android, а также существующие подходы к моделированию пользовательского интерфейса для приложений на базе ОС Android. К существующим подходам относятся: Activity Transition Graph, Window Transition Graph, Screen Transition Graph. Модель SVG состоит из трех основных компонентов: экстрактора топологии, строителя экранов и межкомпонентного анализатора.

Ключевые слова: графический интерфейс пользователя, тестирование, модель генерации, автоматизированное тестирование

REVIEW OF EXISTING APPROACHES TO MODELING THE ANDROID USER INTERFACE

Sarycheva Yu. Yu., Belov Yu. S.

Bauman Moscow State Technical University, Kaluga branch, Kaluga,
e-mail: yulia.sarycheva99@mail.ru

The graphical user interface (GUI) is an integral part of software on most major platforms, including the Android OS, and is of great interest to many researchers. End users interact with the application based on what they want to do with the application and what they see in its GUI. Because different applications often use common user interface design patterns, it is intuitive that the way people interact with a GUI can be generalized across applications. The purpose of an interaction model is to capture such generalized interaction models. Android apps are made up of four main types of components: activities, services, broadcast receivers, and content providers. This study will consider the main entities of the graphical user interface in applications based on the Android OS, as well as existing approaches to modeling the user interface for applications based on the Android OS. Existing approaches include: Activity Transition Graph, Window Transition Graph, Screen Transition Graph. An SVG model consists of 3 main components: a topology extractor, a screen builder, and a cross-component analyzer.

Keywords: graphical user interface, testing, generation model, automated testing

Анализ графического интерфейса приложения представляет большой интерес для многих исследователей и практиков, на данный момент существуют два направления исследований в этой области. Один из них – понять поведение приложений с точки зрения разработки программного обеспечения, другой – с точки зрения взаимодействия человека с компьютером для анализа дизайна пользовательского интерфейса [1].

Цель – исследовать методы, применяющиеся к моделированию GUI для мобильных приложений на базе ОС Android.

Структура Android-приложения

Android – это мобильная операционная система с открытым исходным кодом, в первую очередь предназначенная для мобильных устройств с сенсорными экранами. Впервые она была показана в 2007 г.

ее разработчиками Google и Open Handset Alliance [2]. Самой последней стабильной версией Android является Android 13.

Приложения для Android состоят из четырех основных типов компонентов [3]: действий, сервисов, приемников вещания и поставщиков контента.

Действия – это основные строительные блоки пользовательского интерфейса, которые облегчают взаимодействие между пользователем и приложением.

Активность – это точка входа для взаимодействия с пользователем [4]. Эти действия запускаются, когда приложение использует другое приложение, а также являются точкой взаимодействия с пользователем приложения. Приложения обычно состоят из нескольких таких действий, которые вместе образуют целое приложение, но не связаны друг с другом сильно.

Сервисы выполняют задачи, не требующие пользовательского интерфейса, например предварительную загрузку файлов для более быстрого доступа или воспроизведение музыки в фоновом режиме после того, как пользователь переключился на другие действия.

Широковещательные приемники – это компоненты, которые отвечают на уведомления. Широковещательная рассылка может исходить от:

- 1) системы, например, для объявления о низком заряде батареи;
- 2) другого приложения, например, для уведомления о завершении загрузки файла;
- 3) от других компонентов приложения, например, для уведомления об успешном входе в систему.

Контент-провайдеры управляют доступом к данным, например, для хранения и извлечения контактов.

Существующие подходы

На данный момент существуют следующие подходы к моделированию пользовательского интерфейса Android-приложений:

- ATG – Activity Transition Graph;
- WTG – Window Transition Graph;
- STG – Screen Transition Graph.

График перехода активности (ATG) фиксирует действия приложения и переходы между этими действиями. Он не моделирует другие компоненты приложения и не моделирует действия, которые запускают переходы, т.е. какой пользовательский ввод запускает переход между действиями.

Оконный график перехода (WTG) расширяет ATG, устраняя некоторые из этих ограничений: он моделирует обработчик событий для каждого перехода, а также включает меню и диалоги как отдельные узлы модели. Однако он не рассматривает фрагменты, ящики, службы и широковещательные приемники.

Рассмотрим пример авторизации через Facebook (запрещенная в РФ социальная сеть; принадлежит корпорации Meta,

которая признана в РФ экстремистской и запрещена) в тестовом приложении. Для желаемого результата пользователю необходимо совершить действия, представленные на рис. 1.

Для данного приложения и ATG, и WTG будут включать только три узла, соответствующие действиям приложения: «Контроллер», который соответствует действию на рис. 1, а–с и f, «Аутентификатор» для действия на рис. 1, d, и Facebook (запрещенная в РФ социальная сеть; принадлежит корпорации Meta, которая признана в РФ экстремистской и запрещена) для действия на рис. 1, e, которое реализовано внутри Facebook SDK (запрещенная в РФ социальная сеть; принадлежит корпорации Meta, которая признана в РФ экстремистской и запрещена). Модели ATG и WTG не будут включать переходы, так как все переходы в этом приложении происходят из фрагментов. Таким образом, ни одна из этих моделей не будет содержать путь, ведущий к активности Facebook (запрещенная в РФ социальная сеть; принадлежит корпорации Meta, которая признана в РФ экстремистской и запрещена).

Кроме того, в моделях не учитываются широковещательные приемники, и, следовательно, никакие действия, следующие за входом в Facebook (запрещенная в РФ социальная сеть; принадлежит корпорации Meta, которая признана в РФ экстремистской и запрещена), например, на рис. 1, f, недоступны.

Еще более серьезным недостатком существующих моделей пользовательского интерфейса является то, что они рассматривают компоненты пользовательского интерфейса, такие как действия и меню, как отдельные узлы. Эти модели неточно представляют состав компонентов пользовательского интерфейса на экранах, что препятствует их применимости для сценария исследования, ориентированного на цель: поскольку представления не включают достаточно деталей для извлечения исполняемого пути.

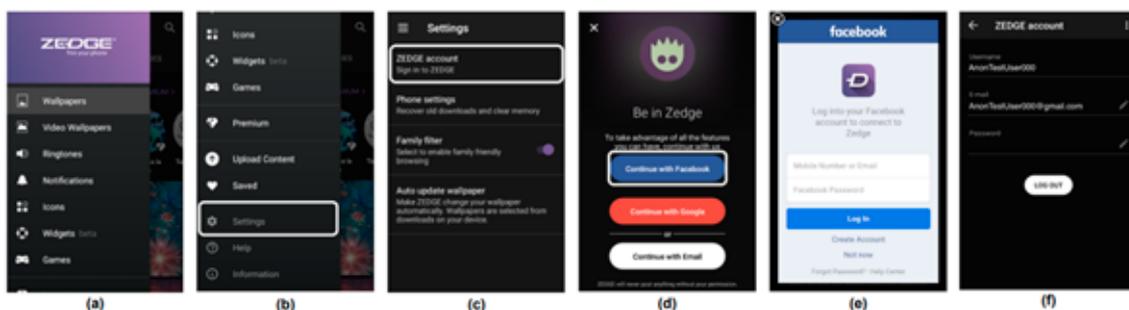


Рис. 1. Пример авторизации



Рис. 2. Экстрактор STG

График перехода экрана (SVG) моделирует экраны приложений, представляя состав активности контейнера, размещенных фрагментов, меню и навигационных диалогов на каждом экране.

В STG основные узлы V_s представляют экраны приложений, а вспомогательные узлы V_r и V_b представляют службы и приемники вещания соответственно. Каждый экраный узел в $V_s \in V_s$ состоит из одного контейнера действий V_sA , нуля или более фрагментов V_sF , нуля или одного меню/ящика V_sM и нуля или одного диалога V_sD . Коллекция всех узлов экрана соответствует возможным экранам, разрешенным в приложении. Действие с n фрагментами и одним ящиком может соответствовать 2×2^n различным экранам (все комбинации фрагментов, с ящиком и без него).

Вспомогательные узлы необходимы, так как переходы между экранами могут проходить через эти элементы. Ребра E в STG соответствуют переходам между узлами $V = V_s \cup V_r \cup V_b$.

Каждое ребро $e \in E$ имеет метку e_t , которая представляет событие, запускающее переход: либо действие пользователя, например нажатие кнопки «Учетная запись» для перехода от элемента № 3 к элементу № 5, либо намерение, которое запускает широковещательный приемник. Переходы, которые запускаются автоматически, например, когда сервис открывает диалог пользовательского интерфейса без каких-либо уведомлений или кликов, не имеют в модели триггеров событий: $e_t = \emptyset$.

Экстрактор STG состоит из трех основных компонентов, показанных на рис. 2: экстрактора топологии, строителя экранов и межкомпонентного анализатора.

Экстрактор топологии

Средство извлечения топологии идентифицирует основные компоненты приложения, т.е. действия, службы и приемники широковещательной рассылки, а также графики их вызовов. Он опирается на реализацию FlowDroid [5] для извлечения компонентов как из исходного кода приложения, так и из XML-файлов и связывания каждого компонента с его точками входа, то есть событиями жизненного цикла и методами

обратного вызова для конкретного приложения. Он также использует FlowDroid для построения графиков вызовов всех точек входа компонента. Например, активность может регистрировать обратные вызовы, которые вызываются при нажатии кнопки или получении обновления местоположения; затем соответствующий обработчик обратного вызова будет связан с графом вызовов действия и проанализирован между событиями жизненного цикла `onResume` и `onPause` действия. Методы приложения, аннотированные «@JavascriptInterface», также могут запускаться веб-представлениями с поддержкой JavaScript [6]. Поскольку вызов этих методов зависит от содержимого WebView, доставляемого во время выполнения, экстрактор топологии консервативно расширяет графы вызовов, предполагая, что любой метод с поддержкой JavaScript в компоненте может быть вызван любым WebView с поддержкой JavaScript в том же компоненте. После извлечения компонентов и их графов вызовов узлы, соответствующие службам и широковещательным приемникам, добавляются непосредственно в STG в V_r и V_b соответственно. Далее строятся экраны приложений в V_s .

Конструктор экрана

Ключевым моментом построения модели является моделирование экранов пользовательского интерфейса и переходов между экранами. С этой целью конструктор экрана анализирует граф вызовов каждого действия, собирая размещенные фрагменты, меню, диалоговые окна и панели навигации. Фрагменты могут быть определены статически в файле макета активности или динамически в коде. Конструктор экрана начинает с извлечения файлов макета с помощью AXMLPrinter [7], а затем идентифицирует объявляемые ими фрагменты. Чтобы собрать объявления фрагментов в коде, конструктор экрана просматривает граф вызовов каждого действия, определяя методы транзакций фрагментов, которые добавляют, удаляют или заменяют фрагменты. Графики активности вызовов сканируются в несколько итераций. В первых, конструктор экрана анализирует

события жизненного цикла действия, инициализированные при инициализации действия: от onCreate до onResume. Для каждого события он идентифицирует методы транзакций фрагментов *m*, которые добавляют, удаляют или заменяют фрагменты; затем он выполняет межпроцедурный, контекстно-зависимый и потокозависимый анализ пути вызова *m*, чтобы идентифицировать тип Java фрагмента (фрагментов), обрабатываемых в *m*. Как только все фрагменты событий жизненного цикла от onCreate до onResume действия *A* обработаны, Screen Builder создает «базовый» экранный узел *Vs* в *Vs* для действия *A*, причем *VsF* содержит (возможно, пустой) список идентифицированных фрагментов. Этот «базовый» экранный узел отображается при запуске действия. Методы жизненного цикла, выдаваемые при приостановке, остановке или возобновлении действия, могут дополнительно добавлять, удалять или заменять фрагменты на «базовом» экране. Таким образом, конструктор экрана анализирует возможные цепочки вызовов методов жизненного цикла: «onPause → onResume» и «onPause → onStop → onRestart → onResume», чтобы извлечь фрагменты из каждой из этих цепочек. Если фрагменты в цепочке отличаются от фрагментов в «базовом» экранном узле *Vs*, добавляется новый экранный узел *Vs'* для активности *A*, которая содержит объединение фрагментов в *Vs* и идентифицированных в цепи. Он также добавляет переход между *Vs* и *Vs'* с пустой меткой, так как этот переход запускается системой автоматически. Точно так же обрабатываются фрагменты в методах обратного вызова. Поскольку порядок обратных вызовов нельзя предсказать, конструктор экрана предполагает, что обратные вызовы могут выполняться в любом порядке; таким образом, он создает новый экран *Vs''*, если он еще не существует, для любого возможного порядка методов обратного вызова, которые изменяют фрагменты «базового» экрана. Конструктор экрана также создает границы перехода между этими экранами и устанавливает метку перехода *et* в качестве действия, запускающего соответствующий обратный вызов. Наконец, конструктор экрана анализирует каждый экран *Vs*, чтобы идентифицировать его меню и диалоги. С этой целью он анализирует графы вызовов активности экрана *VsA* и всех его собранных фрагментов *VsF*, чтобы определить методы, инициализирующие меню, ящики и диалоги. Когда такой метод найден, конструктор экрана копирует экран *Vs* для действия *A* в *Vŝ*, добавляет найденное меню, ящик или диалоговое окно к *Vŝ*, добавляет *Vŝ*

к набору всех экранных узлов *Vs* и создает границу перехода между *Vs* и *Vŝ*. Метка перехода *et* устанавливается как действие, запускающее метод обратного вызова.

Межкомпонентный анализатор

На предыдущем шаге собраны все узлы экрана и переходы между узлами экрана одной активности, но с разными фрагментами, меню, ящиками и диалогами. Межкомпонентный анализатор собирает межкомпонентные переходы между узлами, соответствующими различным компонентам Android, например экранам различных действий, сервисов и приемников вещания. Эти переходы выполняются с помощью методов межкомпонентной связи (ICC), и мы полагаемся на интеграцию FlowDroid с IscTA [7] для определения связей ICC между компонентами приложения. Для каждого канала, где источником и целью являются службы и/или широковещательные приемники, Межкомпонентный анализатор просто создает соответствующий фронт перехода *e* в STG. Если целью перехода является широковещательный приемник, метка перехода *et* устанавливается равной широковещательной рассылке, которая иницирует событие, указанное в фильтре намерений приемника широковещательной рассылки. Если целью перехода является служба, *et* = \emptyset , так как этот переход запускается «автоматически», без какого-либо пользовательского или системного события. Действия представлены несколькими экранами и, таким образом, требуют более тонкого подхода. Если источником связи ICC является действие *A*, межкомпонентный анализатор идентифицирует точку входа действия *p*, из которой исходит сообщение. Затем он находит все узлы экрана в STG, соответствующие *A*, до которых можно добраться из «базового» узла экрана *A* через переходы, связанные с действием, запускающим *p*. Он добавляет переход от всех этих узлов к узлам, представляющим цели, помечая их действием, которое запускает *p*. Когда целью также является действие, межкомпонентный анализатор находит только «базовый» экранный узел этого действия и создает переход к этому узлу. Это связано с тем, что при запуске нового действия оно запускается на начальном экране; переходы между различными экранами целевого действия уже обрабатываются конструктором экрана.

Некоторые действия становятся доступными только после взаимодействия с экраном. Например, только после добавления товара в корзину будет активной кнопка «Оформить заказ». Поскольку STG

не моделирует правильный порядок пользовательских событий (например, щелчков, прокрутки и т.д.) на экране, средство выбора действий динамически взаимодействует с экраном, пытаясь изменить его состояние и активировать требуемое действие. Как и другие инструменты динамического исследования [8], он использует взвешенную стратегию исследования пользовательского интерфейса, которая выбирает следующее событие для конкретного виджета на основе типа события, прошлой частоты выполнения и количества новых виджетов, развертываемых событием.

Если Vs' по-прежнему не может быть достигнуто после определенного количества попыток (в настоящее время установлено 50 итераций), средство выбора действий возвращается и переходит к следующему пути. Если ни на одном из путей, ведущих к текущей цели, нет доступных действий или если еще нет набора «рабочих» целей, средство выбора действий выполняет поиск в ширину, чтобы найти следующую доступную цель, и повторяет поиск этой цели. Он возвращает «unreachable», если не найдено ни одного действия, ведущего к какой-либо из целей.

Заключение

Графический интерфейс является неотъемлемой частью программного обеспечения на большинстве основных платформ, включая Android, и представляет широкий интерес для многих исследователей. Были исследованы методы, применяющиеся к моделированию GUI для мобильных приложений на базе ОС Android: ATG, WTG, STG.

Серьезным недостатком ATG и WTG является то, что они рассматривают компоненты пользовательского интерфейса,

такие как действия и меню, как отдельные узлы. Эти модели неточно представляют состав компонентов пользовательского интерфейса на экранах, что препятствует их применимости для сценария исследования, ориентированного на цель: поскольку представления не включают достаточно деталей для извлечения исполняемого пути. График перехода экрана (SVG) моделирует экраны приложений, представляя состав активности контейнера, размещенных фрагментов, меню и навигационных диалогов на каждом экране, что выгодно отличает его от других методов.

Список литературы

1. Винокуров А.В., Лавлинская О.Ю. Уровни организации автоматизированного тестирования мобильных приложений для операционной системы Android // Вестник Воронежского института высоких технологий. 2020. № 3 (34). С. 22–26.
2. Naja F., Mansur S., Wibawanto A. Automated Software Testing on Mobile Applications: A Review with Special Focus on Android Platform // 20th International Conference on Advances in ICT for Emerging Regions. 2020. P. 4–6.
3. Михалевская К.А., Сергачева М.А. Сравнение инструментов для автоматизации тестирования мобильных приложений на ОС Android // Chronos: естественные и технические науки. 2020. № 2 (30). С. 45–49.
4. Воробьев Н.А., Бурмин Л.Н., Степанов Ю.А. Сравнительный анализ средств тестирования мобильных приложений // Евразийский союз ученых. 2020. № 6–1 (75). С. 36–38.
5. King T.M., Arbon J., Santiago D. AI for Testing Today and Tomorrow: Industry Perspectives. IEEE International Conference On Artificial Intelligence Testing (AITest). 2019. P. 81–88.
6. Сарычева Ю.Ю., Белов Ю.С. Применение искусственного интеллекта в автоматизированном тестировании GUI // Научные исследования в современном мире. Теория и практика: сборник избранных статей Всероссийской (национальной) научно-практической конференции. 2022. С. 55–56.
7. Pan M., Xu To., Pei Yu. GUI-Guided Test Script Repair for Mobile Apps. IEEE Transactions on Software Engineering. 2022. Vol. 48. No. 3. P. 3–5.
8. Плодунин Д.М. Реализация модели автоматизированного тестирования // Огарёв-Online. 2020. № 13 (150).

УДК 004.31:378.147.88

МИКРОПРОГРАММНАЯ РЕАЛИЗАЦИЯ И ЭКСПЕРИМЕНТАЛЬНОЕ ИССЛЕДОВАНИЕ УЧЕБНОГО ПРОЦЕССОРА В MICROSOFT EXCEL

Страбыкин Д.А.

ФГБОУ ВО «Вятский государственный университет», Киров, e-mail: Strabykin@mail.ru

В работе предлагается подход к микропрограммной реализации и экспериментальному исследованию учебного процессора ЭВМ, основанный на применении ранее разработанной функциональной модели базового процессора с микропрограммируемой архитектурой в Microsoft Excel. Приводятся структура и возможности функциональной модели базового процессора. Структура базового процессора с блоком памяти программ и данных отображается на экранной форме, используемой для отладки микропрограмм и программ при проведении экспериментальных исследований и позволяющей наблюдать изменения состояний внутренних регистров процессора. На примере простейшей вычислительной задачи (деление чисел нацело) рассматривается решение задач, связанных с микропрограммной реализацией и экспериментальным исследованием учебного процессора: разработка (выбор) алгоритма решения вычислительной задачи; определение архитектуры процессора и программирование; кодирование программы и распределение памяти программ и данных; разработка микропрограммы командного цикла и составление таблиц прошивок блока памяти микропрограмм и преобразователя кода операции в начальный адрес соответствующей микропрограммы; ввод и отладка микропрограммы командного цикла и программы решения вычислительной задачи. Отмечаются дополнительные возможности функциональной модели базового процессора, упрощающие и ускоряющие отладку микропрограмм и программ: наличие на экранных формах специальных полей, в которых отображаются мнемонические обозначения и пояснения, облегчающие понимание изменений состояний процессора при проведении экспериментальных исследований; использование пультовых дублеров регистров и ячеек памяти, позволяющих заносить данные в регистры и ячейки памяти перед выполнением любой микрокоманды (команды) с использованием клавиатуры; реализация различных режимов работы учебного процессора, обеспечивающих выполнение отдельных микрокоманд, команд и всей программы. Рассмотренный подход способствует более глубокому изучению архитектуры, структуры и рабочего цикла процессора, позволяя обучающемуся разработать простейший учебный процессор, создать и экспериментально исследовать его действующую модель.

Ключевые слова: микропрограммируемые вычислительные структуры, архитектура и структура процессора, действующие функциональные модели, разработка и отладка микропрограмм, компьютерные практикумы по ЭВМ, применение Microsoft Excel

MICROPROGRAM IMPLEMENTATION AND EXPERIMENTAL TEST OF THE TRAINING MICROPROCESSOR IN MICROSOFT EXCEL

Strabykin D.A.

*Federal State Budgetary Educational Institution of Higher Education «Vyatka State University»,
Kirov, e-mail: Strabykin@mail.ru*

This paper suggests a microprogram implementation and experimental tests of the training computer processor based on a previously developed functional model of a basic microprogrammable processor architecture in Microsoft Excel. The structure and capabilities of the functional model of the basic processor are described. The structure of the basic processor with a program and data memory blocks is displayed as a screen form and used to debug microprograms and programs during experimental tests. The screen form allows observing the changes in the states of processor's internal registers. Integer division is used as an example of a simple computational task to demonstrate how a set of problems related to microprogram implementation and experimental test of a training processor is solved. This set of problems includes: development (selection) of an algorithm to solve the computational task, deciding of the processor architecture and programming, coding the program and distributing the memory between programs and data, development of the command cycle microprogram and creation of tables for the microprogram memory block and the converter of operation codes to the initial address of the corresponding microprograms, input and debugging of the command cycle microprogram and the program solving the computational task. The functional model of the basic processor has the following additional features to simplify and accelerate microprogram and program debugging: special fields in the screen form to display mnemonic designations and explanations, which help in understanding processor's state changes during experimental tests; the use of input panels to control register and memory cells, which allow data to be entered with keyboard into registries and memory cells before execution of any microcommand (command); the implementation of various modes of operation of the training processor allowing execution of individual microcommands, commands, and the entire program. The suggested approach helps deeper studies of processor's architecture, structure and operating cycle by allowing the students to design the simplest training processor, implement and experimentally test its functional model.

Keywords: microprogrammable computing devices, processor architecture, structure of processors, functional models, development and debugging of microprograms, practical studies of computers, Microsoft Excel applications

Важную роль при изучении процессоров ЭВМ играют практические занятия и лабораторные работы, позволяющие обучающимся получить более глубокие знания.

Для проведения лабораторных работ часто используются программные модели реальных [1, 2] или учебных [3, 4] процессоров. Программные модели могут быть кросс-

сплатформенными, а также представлять собой веб-приложения. В программных моделях реальных процессоров обучаемым доступна архитектура процессора на уровне системы команд, поэтому экспериментальные исследования в основном связаны с разработкой и покомандным выполнением ассемблерных программ. Программные модели учебных процессоров могут быть ориентированы на микропрограммируемые процессоры и позволять обучающимся проводить самостоятельную разработку микропрограмм интерпретаторов команд и экспериментальное исследование процессоров с различной архитектурой [4, 5]. Такие модели дают возможность исследовать работу процессора не только на уровне архитектуры, но и на уровне «микроархитектуры», отражающей структуру процессора и демонстрирующей потактовое выполнение команд.

Цель исследования: развитие технологии применения функциональных моделей микропрограммируемых вычислительных структур в учебном процессе, использующей в качестве средства моделирования табличный процессор Microsoft Excel и позволяющей осуществлять микропрограммную реализацию и экспериментальное исследование учебного процессора ЭВМ.

Предлагается подход к микропрограммной реализации и экспериментальному исследованию учебного процессора ЭВМ, основанный на применении функциональной модели процессора с микропрограммируемой архитектурой в Microsoft Excel [5]. Моделируемый процессор с микропрограммируемой архитектурой называется далее базовым процессором (БПР), в отличие от учебного процессора (УПР), микропрограммная реализация которого осуществляется с помощью БПР.

*Структура и возможности
функциональной модели базового
процессора*

Используемая функциональная модель, кроме модели восьмиразрядного микропрограммируемого БПР, содержит модель оперативного запоминающего устройства. Структура БПР с запоминающим устройством отображается на экранной форме ПР-Ф при проведении экспериментальных исследований. На экранной форме ПР-Ф присутствуют следующие основные блоки (рис. 1) [5]: БПО – блок пуска-останова; БППД – блок памяти программ и данных; ПНА – преобразователь начального адреса, БПМП – блок памяти микропрограмм; БР – блок регистров; БАЛ – блок арифметико-логический.

Кроме того, в состав базового процессора входят (рис. 1): РК – регистр команд; РС – программный счетчик; РА – регистр адреса БППД, МНА – мультиплексор начального адреса, РА – регистр адреса, СИ – схема инкремента, МА – мультиплексор адреса, «⊕» – инвертор кода условия, МУ – мультиплексор условий и признаков, РУ – регистр условий, РР – регистр признаков, МД – мультиплексор данных, Мс, Мд – мультиплексоры номеров регистров БР; РВ – буферный регистр. На экранной форме ПР-Ф, используемой для проведения экспериментальных исследований (рис. 1), также отображаются: SA – пультовый регистр ввода в РС адреса команды; АМК – пультовый регистр ввода в РА – адреса микрокоманды; ДБР – пультовый дублер блока регистров.

Основу операционной части базового процессора составляют БР и БАЛ. БР содержит восемь восьмиразрядных регистров. Содержимое любой пары регистров с помощью двух трехразрядных входов номеров регистров может быть выбрано и передано на входы БАЛ, в котором выполняется одна из восьми микроопераций и формируются значения трех логических условий. Управляющая часть БПР представляет собой устройство управления с программируемой логикой, в памяти которого хранятся двоичные коды микрокоманд. На рисунке 2 показаны назначение и значения кодов многоразрядных полей микрокоманды (где V – наименование, n – разрядность поля, а $M[A]$ – содержимое ячейки памяти с адресом A).

БПМП отображается на экранной форме проведения исследований ПР-Ф в виде блока памяти, в котором выделено окно микрокоманд, содержащее восемь последовательно расположенных 33-разрядных ячеек памяти (рис. 1а). Содержимое окна обновляется после выполнения каждой микрокоманды. Занесение данных в ПНА и БПМП производится на листе ПР-В путем ввода данных с клавиатуры в ячейки таблиц, моделирующих ПНА и БПМП, а также ячейки, предусмотренные для ввода дополнительной информации (пояснения, коды операций).

Работа БППД моделируется с помощью таблицы, расположенной на листе ПР-В. Ввод данных в БППД производится на экранной форме ПР-В аналогично записи в БР с помощью ячеек пультового дублера БППД. Данные вводятся с клавиатуры в выбранные дублирующие регистры БР (ячейки БППД), а соответствующие им управляющие сигналы v (vm) устанавливаются в единичное. При нажатии клавиши F9 данные из дублирующих регистров (ячеек), у которых $v=1$ ($vm=1$), переносятся в основные регистры БР (БППД).

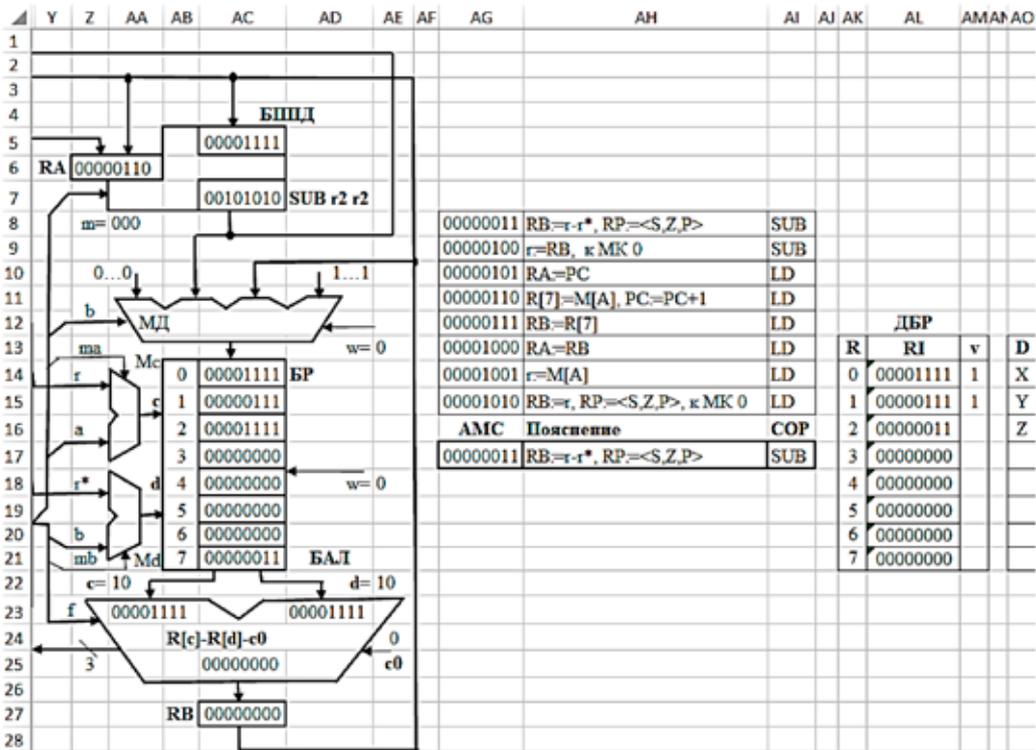
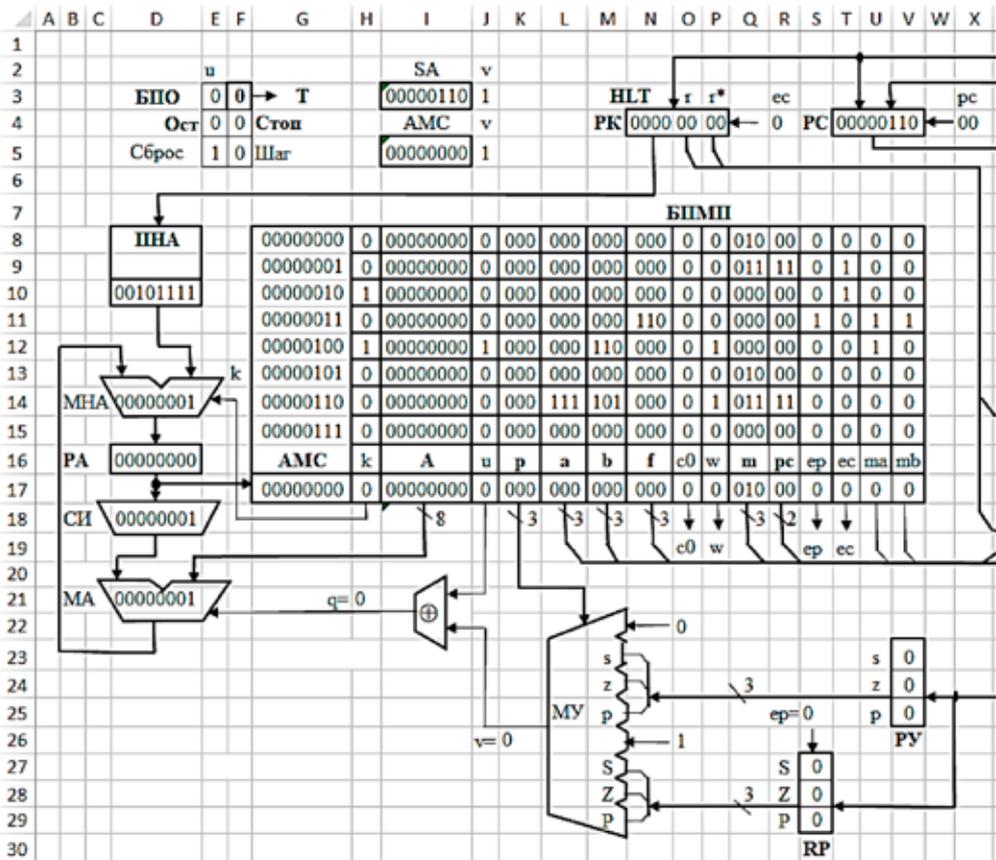


Рис. 1. Экранная форма ПР-Ф: левая часть (а) и правая часть (б)

V	n	Назначение
k	1	Занесение в РА микрокоманды адреса из ПНА (при k=1)
A	8	Адрес следующей микрокоманды
u	1	Инверсия значения выбранного условия (признака) (при u=1)
p	3	Код выбираемого условия (признака)
a	3	Номер регистра в БР (при чтении подается на левый вход БАЛ)
b	3	Номер регистра в БР (при чтении подается на правый вход БАЛ)
f	3	Код микрооперации, выполняемой в БАЛ
c0	1	Значение входа переноса в младший разряд БАЛ
w	1	Запись в БР (при w=1)
m	3	Код микрооперации, выполняемой в БППД и РА
pc	2	Код микрооперации, выполняемой в РС
ep	1	Запись в регистр признаков RP (при ep=1)
ec	1	Запись в регистр команд РК (при ec=1)
ma	1	Выбор источника номера регистра (c=a при ma=0 и c=0r при ma=1)
mb	1	Выбор источника номера регистра (d=b при mb=0 и d=0r* при mb=1)

а)

f	Микрооперация
000	RB=R[c]
001	RB=R[c]&R[d]
010	RB=R[c]∨R[d]
011	RB=R[c]⊕R[d]
100	RB=2R[d].0
101	RB=0.R[d]/2
110	RB=R[c]-R[d]-c0
111	RB=R[c]+R[d]+c0

б)

p	Признак
000	0
001	s
010	z
011	p
100	1
101	S
110	Z
111	P

в)

w	b	БР
0	xxx	Хранение
1	100	R[c]:=00000000
1	101	R[c]:=M[A]
1	110	R[c]:=RB
1	111	R[c]:=11111111

г)

m	БППД	РА
000	Хранение	Хранение
001	Хранение	RA:=RB
010	Хранение	RA:=PC
011	Чтение	Хранение
100	Запись	Хранение

д)

pc	РС
00	Хранение
01	PC:=RB
10	PC:=M[A]
11	PC:=PC+1

е)

Рис. 2. Назначение полей микрокоманды (а) и значения кодов многоразрядных полей микрокоманды: БАЛ (б), БР (в), РС (г), БППД и РА (д), признаков (е)

Кроме того, по сигналу «Сброс» при единичном значении соответствующего управляющего сигнала (v) в РС (РА) может быть занесен предварительно набранный на клавиатуре стартовый адрес команды SA (адрес микрокоманды АМК).

Процессор функционирует под действием сигналов, вырабатываемых блоком пуска-останова БПО (рис. 1а), в состав которого входит генератор тактовых импульсов (ГТИ). В БПО с клавиатуры подаются

сигналы «Сброс» (E5) и «Шаг» (F5) (в соответствующую ячейку MS Excel вводится необходимое значение сигнала (0 или 1) и нажимается клавиша F9). В БПО вырабатываются два признака: «Ост» и «Стоп». Признак «Ост» (останов) принимает единичное значение, если выполняется микрокоманда, содержащая микрооперацию останова (p=100), и нулевое значение – по сигналу «Сброс». Признак «Стоп» принимает единичное значение, если единичное значение

имеет сигнал «Шаг» и выполняется микрокоманда, содержащая микрооперацию ($k=1$) перехода по коду операции, и нулевое значение – по сигналу «Сброс».

Проведению экспериментальных исследований предшествует изменение параметров Microsoft Excel, связанных с вычислением формул (выбираются вычисления в книге «вручную» и включаются итеративные вычисления с предельным числом итераций, превышающим удвоенное максимальное число микрокоманд, которые необходимо выполнить после нажатия клавиши F9). БПО обеспечивает работу БПР в следующих режимах: выполнение микрокоманды за два полутакта; выполнение микрокоманды за один такт; выполнение команды; выполнение программы.

Применение функциональной модели базового процессора в учебном процессе

Функциональная модель базового процессора позволяет производить микропрограммную реализацию и экспериментальное исследование учебных процессоров с различной архитектурой. В качестве исходных данных может выступать простая задача, на примере программного решения которой исследуется работа учебного процессора. Применение функциональной модели базового процессора позволяет обучающимся решать следующие проектные задачи для учебного процессора.

1. Разработка (выбор) алгоритма решения задачи.
2. Определение архитектуры процессора и программирование.
3. Кодирование программы и распределение памяти программ и данных.
4. Разработка микропрограммы командного цикла и составление таблиц прошивок ПНА и БПМП.
5. Ввод и отладка микропрограммы командного цикла и программы решения задачи.

Рассмотрим применение функциональной модели БПР в учебном процессе

на примере операции деления нацело 8-разрядного двоичного числа X на 8-разрядное двоичное число Y .

1. Разработка (выбор) алгоритма решения задачи. Для выполнения заданной операции предлагается использовать следующий алгоритм. Если число $Y=0$, то деление не выполняется, а формируется признак переполнения P . В противном случае из делимого X вычитается делитель Y , и если разность R положительное число, то частное Z увеличивается на единицу, а разность R выступает в качестве нового уменьшаемого на следующем шаге. Процесс вычитания продолжается до тех пор, пока не будет получена отрицательная разность. В этом случае Z будет представлять собой искомое частное от деления X на Y нацело [МПВУ].

2. Определение архитектуры УПР и программирование. Определение архитектуры включает выбор программистской структуры (состава программно-доступных регистров) и определение внутреннего языка УПР (системы данных и системы команд). Программистская структура содержит: четыре общих регистра ($r0-r3$), программный счетчик (PC) и регистр признаков (RP), включающий три разряда: знака (S), нуля (Z) и переполнения (P) (рис. 3а).

Разработка системы команд предполагает определение набора операций, способов адресации, модификаций и форматов команд. Для рассматриваемого примера система команд УПР приведена в таблице 1, а форматы команд – на рисунке 3б. В таблице 1 использованы следующие обозначения: $r, r^* \in \{r0, r1, r2, r3\}$ – программно-доступные регистры: регистр r^* является источником данных, а регистр r – приемником результата, но может также служить источником второго операнда; $M[A]$ – ячейка памяти с адресом A ; знак «+» в описании признаков означает, что устанавливается новое значение признака по результату выполнения команды, а знак «-» свидетельствует о сохранении старого значения признака.

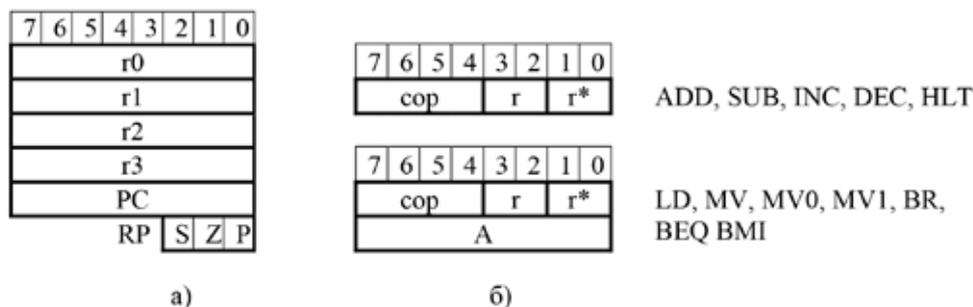


Рис. 3. Регистры программистской структуры (а) и форматы команд (б)

Таблица 1

Система команд

Наименование	Мнемоника	Описание	Признаки		
			S	Z	P
СУММИРОВАНИЕ	ADD r r*	r:=r+r*, PC:=PC+1	+	+	+
ВЫЧИТАНИЕ	SUB r r*	r:=r-r*, PC:=PC+1	+	+	+
Инкремент	INC r	r:=r+1, PC:=PC+1	+	+	+
Декремент	DEC r	r:=r-1, PC:=PC+1	+	+	+
ЧТЕНИЕ В РЕГИСТР	LD r A	r:=M[A], PC:=PC+2	+	+	+
ЗАПИСЬ ИЗ РЕГИСТРА	MV r A	M[A]:=r, PC:=PC+2	+	+	+
Запись НУЛЕЙ	MV0 A	M[A]:=00000000, PC:=PC+2	0	1	0
Запись ЕДИНИЦ	MV1 A	M[A]:=11111111, PC:=PC+2	1	0	0
ПЕРЕХОД	BR A	PC:=A	-	-	
ПЕРЕХОД, ЕСЛИ НУЛЬ	BEQ A	Если PZ=1, то PC:=A, иначе PC:=PC+2	-	-	
ПЕРЕХОД, ЕСЛИ МИНУС	BMI A	Если PS=1, то PC:=A, иначе PC:=PC+2	-	-	-
ОСТАНОВ	HLT	Останов	-	-	-

SUB r2 r2	Очистка регистра r2 частного Z
LD r1 AY	Загрузка в r1 делителя Y
BEQ m1	Если признак Z=1 (Y=0), то переход на метку m1
LD r0 AX	Загрузка в r0 делимого X
m3 SUB r0 r1	Вычитание из делимого X делителя Y
BMI m2	Если признак S=1, то переход на метку m2
INC r2	Увеличение на единицу частного Z
BR m3	Переход на метку m3
m1 MV1 AP	Запись единиц в ячейку признака P
BR m4	Переход на метку m4
m2 MV0 AP	Запись нулей в ячейку признака P
MV r2 AZ	Запись частного из регистра r2 в ячейку частного Z
m4 HLT	Останов

Рис. 4. Программа деления чисел нацело

Программа деления чисел нацело с использованием символики ассемблера, составленная в предположении, что исходные данные и результаты размещаются в блоке памяти программ и данных, приведена на рисунке 4.

3. Кодирование программы и распределение памяти программ и данных. Двоичные коды операций и начальные адреса их микропрограмм в блоке памяти микропрограмм приведены в таблице 2.

Ввод данных и команд в БППД производится с помощью экранной формы ввода ПР-В. Двоичные коды данных и команд для рассматриваемого примера занимают 28 ячеек памяти. Распределение памяти программ и данных для первых 23 ячеек показано на рисунке 5а.

Таблица 2

Коды операции и начальные адреса микропрограмм операций

Мнемоника	Код операции	Начальный адрес МП
HLT	0000	00101111
ADD	0001	00101111
SUB	0010	00000011
INC	0011	00001110
DEC	0100	00010100
LD	1000	00000101
MV	1001	00100111
BR	1010	00010111
BEQ	1011	00001011
BMI	1100	00010001
MV1	1101	00011001
MV0	1110	00100000

▲	Z	AA	AB	AC	AD	AE	AF	AG	AH		
1											
2		БПД									
3											
4	N	АЯП	D	DI	vm	Мнемоника	Комментарий				
5	0	00000000	00000110	00000110	1	SA	Начальный адрес программы				
6	1	00000001	00000111	00000111	1	X	Делимое				
7	2	00000010	00000111	00000111	1	Y	Делитель				
8	3	00000011	00010000	00000000		Z	Частное				
9	4	00000100	00000000	01010101		P	Признак переполнения				
10	5	00000101	00000000	00000000	1		Свободная ячейка памяти (ЯП)				
11	6	00000110	00101010	00101010	1	SUB r2 r2	Очистка регистра r2 частного Z				
12	7	00000111	10000100	10000100	1	LD r1	Загрузка в r1 делителя Y из ЯП с адресом AY				
13	8	00001000	00000010	00000010	1	AY					
14	9	00001001	10110000	10110000	1	BEQ m1	Если PZ=1 (Y=0), то переход на метку m1				
15	10	00001010	00010011	00010011	1	Am1					
16	11	00001011	10000000	10000000	1	LD r0	Загрузка в r0 делимого X из ЯП с адресом AX				
17	12	00001100	00000001	00000001	1	AX					
18	13	00001101	00100001	00100001	1	SUB r0 r1	Вычитание из делимого X делителя Y				
19	14	00001110	11000000	11000000	1	BMI m2	Если PS=1, то переход на метку m2				
20	15	00001111	00010111	00010111	1	Am2					
21	16	00010000	00111000	00111000	1	INC r0	Увеличение на единицу частного Z				
22	17	00010001	10100000	10100000	1	BRm3	Переход на метку m3				
23	18	00010010	00001101	00001101	1	Am3					
24	19	00010011	11010000	11010000	1	MV1	Запись единиц в ЯП признака P				
25	20	00010100	00000100	00000100	1	AP					
26	21	00010101	10100000	10100000	1	BRm4	Переход на метку m4				
27	22	00010110	00011011	00011011	1	Am4					
28	23	00010111	11100000	11100000	1	MV0	Запись нулей в ЯП признака P				

а)

▲	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
1																			
2		БМПД																	
3																			
4	N	AMC	k	A	u	p	a	b	f	c0	w	m	pc	ep	ec	ma	mb	Пояснение	
5	0	00000000	0	00000000	0	000	000	000	000	000	0	0	010	00	0	0	0	RA=PC	COP
6	1	00000001	0	00000000	0	000	000	000	000	000	0	0	011	11	0	1	0	PC=M[A], PC=PC+1	RC
7	2	00000010	1	00000000	0	000	000	000	000	000	0	0	000	00	0	0	0	Переход по КОП	RC
8	3	00000011	0	00000000	0	000	000	000	110	000	0	0	000	00	1	0	1	RB=r-r*, RP=<S,Z,P>	SUB
9	4	00000100	0	00000000	1	000	000	110	000	000	0	1	000	00	0	0	1	r=RB, к МК 0	SUB
10	5	00000101	0	00000000	0	000	000	000	000	000	0	0	010	00	0	0	0	RA=PC	LD
11	6	00000110	0	00000000	0	000	111	101	000	000	0	1	011	11	0	0	0	R[7]=M[A], PC=PC+1	LD
12	7	00000111	0	00000000	0	000	111	000	000	000	0	0	000	00	0	0	0	RB=R[7]	LD
13	8	00001000	0	00000000	0	000	000	000	000	000	0	0	001	00	0	0	0	RA=RB	LD
14	9	00001001	0	00000000	0	000	000	101	000	000	0	1	011	00	0	0	1	r=M[A]	LD
15	10	00001010	0	00000000	1	000	000	000	000	000	0	0	000	00	1	0	1	RB=r, RP=<S,Z,P>, к МК 0	LD
16	11	00001011	0	00001101	0	110	000	000	000	000	0	0	010	00	0	0	0	RA=PC, при Z=1, к МК 1101	BEQ
17	12	00001100	0	00000000	1	000	000	000	000	000	0	0	000	11	0	0	0	PC=PC+1, к МК 0	BEQ
18	13	00001101	0	00000000	1	000	000	000	000	000	0	0	011	10	0	0	0	PC=M[A], к МК 0	BEQ
19	14	00001110	0	00000000	0	000	111	100	000	000	0	1	000	00	0	0	0	R[7]=0	INC
20	15	00001111	0	00000000	0	000	000	111	111	1	0	000	00	1	0	1	0	RB=r+R[7]+1, RP=<S,Z,P>	INC
21	16	00010000	0	00000000	1	000	000	110	000	000	0	1	000	00	0	0	1	r=RB, к МК 0	INC
22	17	00010001	0	00010011	0	101	000	000	000	000	0	0	010	00	0	0	0	RA=PC, при S=1 к МК 10011	BMI
23	18	00010010	0	00000000	1	000	000	000	000	000	0	0	000	11	0	0	0	PC=PC+1, к МК 0	BMI
24	19	00010011	0	00000000	1	000	000	000	000	000	0	0	000	10	0	0	0	PC=M[A], к МК 0	BMI

б)

Рис. 5. Экранная форма ввода ПР-В: область ввода данных и программ в БПД (а), область ввода микропрограмм в БМПД (б)

4. Разработка микропрограммы командного цикла и составление таблиц прошивок ПНА и БПМП. При разработке микропрограммы командного цикла (интерпретатора команд) производится распределение регистров УПР. В примере регистры программистской структуры r0–r3 занимают в блоке регистров регистры R[0]–R[3], программный счетчик и регистр признаков помещаются в одноименные регистры: PC и RP соответственно. Также в качестве вспомогательного, не являющегося регистром программистской структуры, используется регистр R[7] блока регистров. Микропрограмма командного цикла состоит из 48 микрокоманд. В первых трех ячейках БПМП находятся микрокоманды, которые обеспечивают: пересылку адреса команды из программного счетчика в регистр адреса БППД; чтение команды из БППД в регистр команд и увеличение содержимого программного счетчика на единицу; переход по коду операции к начальному адресу соответствующей микропрограммы (загрузку в RA из ПНА адреса первой микрокоманды микропрограммы выбранной операции). В остальных 45 ячейках размещаются микропрограммы 12 операций в соответствии с системой команд УПР. Ввод микропрограмм в БПМП производится с помощью экранной формы ввода ПР-В. Двоичные коды микрокоманд для рассматриваемого примера для первых 20 ячеек показаны на рисунке 5б.

5. Ввод и отладка микропрограммы командного цикла и программы решения задачи. Ввод данных осуществляется с помощью экранной формы ввода ПР-В. Для каждой операции в ячейку памяти ПНА, адрес которой совпадает с кодом операции, с клавиатуры заносится двоичный код начального адреса микропрограммы этой операции. В экранной форме ПР-В слева от каждой ячейки ПНА предусмотрено дополнительное поле для ввода мнемонического обозначения введенного кода операции. Это мнемоническое обозначение отображается на экранной форме исследования ПР-Ф над регистром команд при загрузке в него команды с данным кодом операции.

Микропрограмма командного цикла вводится в БПМП путем занесения с клавиатуры двоичных кодов микрокоманд. В экранной форме ПР-В справа от каждой ячейки БПМП предусмотрены два вспомогательных поля: «Пояснение» (для ввода описания выполняемых в микрокоманде микроопераций) и «СОР» (для ввода мнемонического обозначения подмикропрограммы, к которой относится данная микрокоманда).

Отладка микропрограммы командного цикла осуществляется с помощью экранной формы исследования ПР-Ф в режимах выполнения микрокоманд по полутактам и тактам. Для упрощения отладки на экранной форме ПР-Ф при выполнении микрокоманд отображается следующая дополнительная информация: источник адреса при занесении адреса в регистр адреса БППД (RA:=PC или RA:=RB); двоичный код данного, поступающий с выбранного входа мультиплексора данных для записи в БР; двоичные коды номеров регистров и находящихся там данных, поступающие на входы БАЛ для выполнения микрооперации; символическое обозначение выполняемой в БАЛ микрооперации; содержание вспомогательных полей микрокоманд («Пояснение» и «СОР») в продолжении окна микрокоманд. Кроме того, значения управляющих сигналов из микрокоманды дублируются на управляющих входах узлов и блоков, на которые они подаются. В процессе отладки микропрограммы командного цикла осуществляется выполнение необходимых последовательностей микрокоманд путем занесения в регистр адреса БПМП начального адреса первой микрокоманды, который предварительно вводится с клавиатуры в поле АМК. Перед выполнением любой микрокоманды также возможно изменение содержимого выбранных регистров в БР с помощью пультового дублера БР.

Ввод двоичных кодов команд и исходных данных программы в БППД производится в форме ввода ПР-В с помощью ячеек пультового дублера БППД. Команда может занимать одну или две ячейки БППД. В экранной форме ПР-В справа от каждой ячейки БППД предусмотрены два вспомогательных поля: «Мнемоника» (для ввода символического обозначения команд и данных) и «Комментарий» (для ввода описания выполняемой командой операции или данного).

Отладка программы решения задачи производится с помощью экранной формы исследования ПР-Ф в режиме выполнения программы по командам. При этом биты *vm* в БППД на форме ввода ПР-В перед выполнением программы устанавливаются в нулевое состояние. Для упрощения отладки на экранной форме исследования ПР-Ф при выполнении команд на выходе БППД отображается символическое обозначение из поля «Мнемоника» формы ввода ПР-В для считываемого содержимого ячейки памяти.

В процессе отладки программы выполняются отдельные команды и необходимые последовательности команд путем занесе-

ния в программный счетчик РС начального адреса первой команды, который предварительно вводится с клавиатуры в поле SA. Перед выполнением любой команды также возможно изменение содержимого выбранных регистров в БР (ячеек БППД) с помощью пульта дублера БР (БППД).

После завершения отладки микропрограммы командного цикла и программы решения задачи работа УПР исследуется в режиме выполнения программы. При этом изменяются параметры Microsoft Excel, связанные с вычислением формул: выбираются вычисления в книге «вручную» и включаются итеративные вычисления с предельным числом итераций, превышающим удвоенное максимальное число микрокоманд, необходимых для выполнения программы. В процессе исследования программы могут быть получены результаты решения задачи, подтверждающие корректность выполнения программы по всем ветвям алгоритма.

Заключение

В применяемой функциональной модели процессора с микропрограммируемой архитектурой, кроме базовых функций, необходимых для реализации учебных процессоров, предусмотрены следующие дополнительные возможности:

- наличие на экранных формах, предназначенных для экспериментального исследования и ввода информации, специальных полей, в которых отображаются мнемонические обозначения и пояснения, облегчающие понимание изменений состояний процессора при проведении экспериментальных исследований;

- использование пультовых дублеров регистров и ячеек памяти, позволяющих заносить данные в регистры и ячейки памяти перед выполнением любой микрокоманды (команды) с использованием клавиатуры, что упрощает и ускоряет отладку микропрограмм и программ;

- наличие различных режимов работы учебного процессора, позволяющих выполнять отдельные микрокоманды, команды и всю программу.

Микропрограммная реализация и экспериментальное исследование учебных процессоров осуществляются по индивидуальным заданиям и способствуют более глубокому изучению архитектуры, структуры и рабочего цикла процессора, позволяют разработать простейший учебный процессор, создать и исследовать его действующую модель. При наличии у обучающегося необходимой предварительной теоретической подготовки решение задачи построения процессора в рассмотренной постановке не вызывает больших затруднений и формирует целостное представление о процессоре. Поскольку экспериментальные исследования выполняются в табличном процессоре Microsoft Excel, с которым, как правило, обучающиеся уже знакомы, им не требуется дополнительного времени для изучения программного интерфейса.

Список литературы

1. CPULator Simulator Documentation. URL: https://cpulator.01xz.net/doc/#sim_processors (дата обращения: 31.03.2023).
2. Титовский С.Н., Титовская Н.В. Учебный программный эмулятор процессора 18086 // Наука и современность. 2014. № 27. С. 192-196.
3. Молчанов О.Е., Васильев А.С., Белая Т.И. Компьютерный тренажер-эмулятор учебной цифровой вычислительной машины // Инженерный вестник Дона. 2015. № 2-2 (36). С. 110.
4. Коннов Н.Н., Пчелинцев А.И., Федулов В.Д., Бычков А.С. Учебный кроссплатформенный эмулятор микропрограммируемой ЭВМ // Информатика: проблемы, методы, технологии: материалы XXII Международной научно-практической конференции им. Э.К. Алгазина (Воронеж, 10-12 февраля 2022 г.) / Под ред. Д.Н. Борисова. Воронеж: Издательство ООО «Вэлборн», 2022. С. 1403-1411.
5. Страбыкин Д.А. Функциональная модель учебного процессора с микропрограммируемой архитектурой в Microsoft Excel // Современные наукоемкие технологии. 2023. № 2. С. 102-120.

УДК 004.942

АЛГОРИТМ КОНТРОЛЯ ВЫПОЛНЕНИЯ ЛОГИЧЕСКИХ ОПЕРАЦИЙ В ПРОГРАММИРУЕМЫХ ЛОГИЧЕСКИХ ИНТЕГРАЛЬНЫХ СХЕМАХ, ВЫПОЛНЯЮЩИХ КРИПТОГРАФИЧЕСКОЕ ПРЕОБРАЗОВАНИЕ ИНФОРМАЦИИ

Лукин М.В.

*Научно-исследовательский центр Центрального научно-исследовательского института
Военно-воздушных сил (Министерства обороны Российской Федерации),
Щелково, e-mail: maxim@vishnin.ru*

В статье приведены разработанные алгоритмы контроля выполнения логических операций (сложение по модулю 2, сложение по модулю 2 в 16-й степени, операция сдвига и подстановки) в программируемых логических интегральных схемах, используемых в качестве основных вычислительных компонентов (логических вентилей) в устройствах, осуществляющих криптографическое преобразование входной информации. Данные логические операции являются основными преобразованиями в криптографии. Преобразование осуществляется с использованием блочного симметричного шифра, в основе которого лежат ключ и псевдослучайная последовательность, называемая синхросвязкой. Данные алгоритмы позволяют блоку управления автономным техническим средством определять правильность выполнения одних из основных операций криптографического преобразования информации с целью выявления и, по возможности, устранения сбоев в работе. Благодаря использованию указанных алгоритмов повышаются такие важные характеристики, как надежность и достоверность. При использовании средних по производительности программируемых логических интегральных схем ущерб в скорости обработки входной информации сводится к минимуму, что позволяет функционировать автономному техническому средству без каких-либо задержек, но с более высокой эффективностью. Данные алгоритмы наиболее актуально использовать в автономных технических средствах, осуществляющих передачу телеметрической информации по радиоканалам с большого удаления по расстоянию от оператора автономного технического средства.

Ключевые слова: программируемая логическая интегральная схема, криптографическое преобразование информации, автономное техническое средство, алгоритм контроля, сложение по модулю

ALGORITHM FOR CONTROLLING THE EXECUTION OF LOGICAL OPERATIONS IN PROGRAMMABLE LOGIC INTEGRATED CIRCUITS PERFORMING CRYPTOGRAPHIC TRANSFORMATION OF INFORMATION

Lukin M.V.

*Research Center of the Central Research Institute of Air Force
(Ministry of Defense of the Russian Federation), Shchelkovo, e-mail: maxim@vishnin.ru*

The article presents the developed algorithms for controlling the execution of logical operations (addition modulo 2, addition modulo 2 to the 16th degree, shift and substitution operation) in programmable logic integrated circuits used as the main computational components (logic gates) in devices performing cryptographic transformation of input information. These logical operations are the main transformations in cryptography. The conversion is carried out using a block symmetric cipher, which is based on a key and a pseudorandom sequence called a synchro link. These algorithms allow the control unit of an autonomous technical means to determine the correctness of performing one of the basic operations of cryptographic transformation of information in order to identify and, if possible, eliminate malfunctions. Thanks to the use of these algorithms, such important characteristics as reliability and reliability are increased. When using medium-performance programmable logic integrated circuits, the damage in the processing speed of input information is minimized, which allows the autonomous technical means to function without any delays, but with higher efficiency. These algorithms are most relevant to use in autonomous technical means that transmit telemetry information over radio channels from a great distance away from the operator of an autonomous technical means.

Keywords: programmable logic integrated circuit, cryptographic transformation of information, autonomous technical means, control algorithm, addition modulo

Криптографическим преобразованием информации (далее – КПИ) называется процесс изменения информации, зависящий от изменяемого параметра и обладающий свойством невозможности восстановления исходной информации по преобразованной, без знания действующего ключа, с трудоемкостью меньше заданной [1].

В настоящее время КПИ осуществляется программными, аппаратными и программно-аппаратными средствами. При этом в

случае даже минимальной ошибки в 1 бит процесс восстановления информации становится невозможным.

Особенно актуальна проблема возникновения ошибки в ходе КПИ на автономных технических средствах (далее – АТС), осуществляющих работу на большом расстоянии от технического оператора и передающих криптографически преобразованную информацию на объект обработки по радиоканалам [2].

В связи с этим возникает необходимость автономного контроля выполнения КПИ на АТС для заблаговременного определения ошибки и, по возможности, устранения сбоя в работе устройства, осуществляющего КПИ.

Основным вычислительным устройством, выполняющим КПИ, в настоящее время служит программируемая логическая интегральная схема. Данное вычислительное средство является наиболее универсальным для реализации различных алгоритмов КПИ.

В основе наиболее криптостойких алгоритмов лежат четыре основные логические операции [3]:

- операция сложения по модулю 2;
- операция сложения по модулю 2^{16} ;
- операция циклического сдвига;
- операция подстановки.

Контроль данных логических операций позволит наиболее полно определять техническое состояние ПЛИС, осуществляемой КПИ на АТС.

Рассмотрим числовой контроль логических операций. В основе построения схем контроля лежат две теоремы [4, 5].

Теорема 1. Сумма чисел сравнима по модулю q с суммой остатков r этих же чисел, то есть:

$$\sum_{i=1}^n A_i = \sum_{i=1}^n r_{ai} \pmod q \quad (1)$$

Теорема 2. Произведение чисел сравнимо по модулю q с произведением остатков r этих же чисел, то есть:

$$\prod_{i=1}^n A_i = \prod_{i=1}^n r_{ai} \pmod q \quad (2)$$

Рассмотрим схему контроля для логической операции сложения по модулю 2.

Схема контроля преобразования [6] (суммирования) двух чисел A_1 и A_2 по модулю два представлена на рисунке 1.

Поясним работу данной схемы следующим образом [7, 8]. Результатом суммирования двух чисел A_1 и A_2 есть число A_3 . В блоках B_1 и B_2 осуществляется вычисление остатков r_1 и r_2 чисел A_1 и A_2 до преобразования соответственно. Далее в блоке S происходит суммирование полученных остатков. Поскольку сумма остатков может быть больше модуля, то на выходе сумматора необходимо еще раз выполнить операцию нахождения остатка с помощью преобразования в блоке B_3 . В результате выполняется сравнение остатка r_3 , полученного от числа A_3 , с суммой остатков чисел A_1 и A_2 – r_Σ с формированием признака «норма» (P) в виде логической единицы и нуля в противном случае.

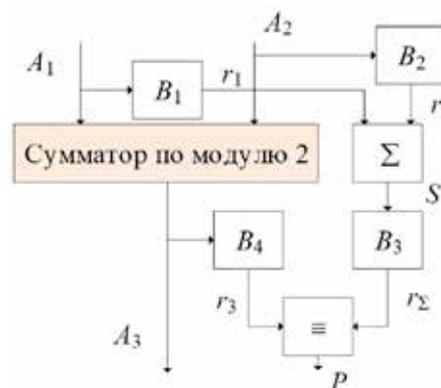


Рис. 1. Схема контроля суммирования по модулю 2

Алгоритм функционирования схемы контроля операции сложения по модулю 2 представлен на рисунке 2.

Условно работу данного алгоритма можно разбить на несколько этапов.

1. На начальном этапе осуществляются получение чисел A_1 и A_2 , участвующих в операции сложения по модулю 2, а также получение остатков r_1 и r_2 из данных чисел.

2. На следующем этапе выполняются операция сложения по модулю 2 чисел A_1 и A_2 (получение числа A_3), а также суммирование остатков r_1 и r_2 ($S = r_1 + r_2$).

3. На третьем этапе осуществляется поиск остатков от результата суммирования S и A_3 .

4. На заключительном этапе осуществляется сравнение остатков от суммы чисел A_1 и A_2 (числа A_3) и суммы остатков r_1 и r_2 (S).

С формальной точки зрения данные этапы могут быть представлены совокупностью отображений:

$$B_1 : A_1 \rightarrow r_1, \quad (3)$$

где B_1 – оператор, характеризующий процедуру получения остатка r_1 от числа A_1 ;

$$B_2 : A_2 \rightarrow r_2, \quad (4)$$

где B_2 – оператор, характеризующий процедуру получения остатка r_2 от числа A_2 ;

Необходимо отметить, что выражения (3) и (4) являются эквивалентными по функциональному представлению операторов B_1 и B_2 .

$$S : \sum_{i=1}^2 r_i \rightarrow c, \quad (5)$$

где S – оператор, формализующий нахождение суммы остатков слагаемых r_1 и r_2 , полученных в результате отображений (3) и (4);

$$B_3 : c \rightarrow r_\Sigma, \quad (6)$$

где B_3 – оператор, характеризующий процедуру получения остатка r_Σ от суммы остатков r_1 и r_2 ;

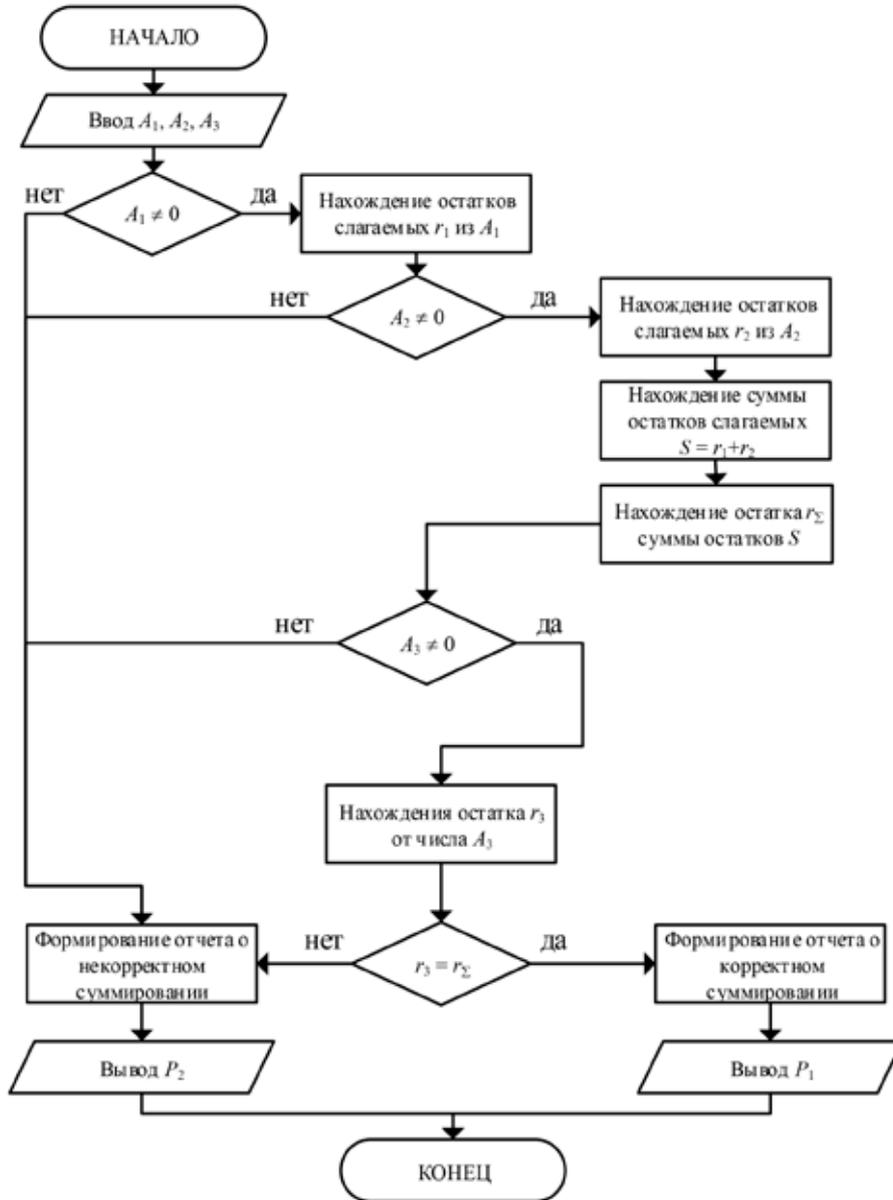


Рис. 2. Алгоритм выполнения контроля операции сложения по модулю 2

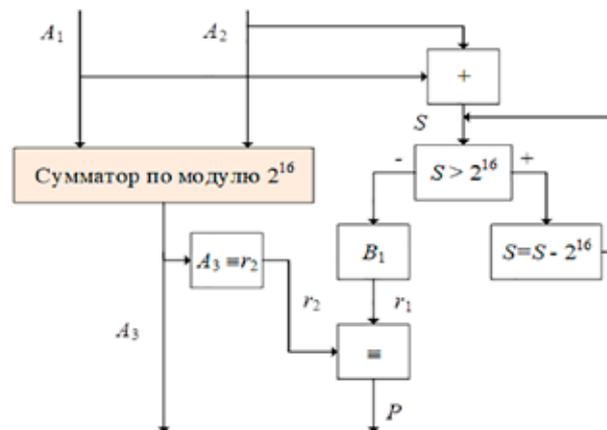


Рис. 3. Схема контроля суммирования по модулю 2^{16}

$$B_4 : A_3 \rightarrow r_3, \quad (7)$$

где B_4 – оператор, характеризующий процедуру получения остатка r_3 от суммы чисел A_1 и A_2 (числа A_3);

$$\alpha : r_3 \vee r_\Sigma \rightarrow p, p = \{0,1\}, \quad (8)$$

где α – оператор, формализующий процедуру сравнения r_3 и r_Σ .

При контроле сложения по модулю 2^{16} схема упрощается (рис. 3) с учетом того, что вычисляется остаток двух чисел в са-

мом преобразовании. Необходимо отметить, что в данном случае результатом выполнения суммирования по модулю 2^{16} является получение непосредственно остатков чисел, что с практической точки зрения позволяет упростить реализацию процесса контроля. В свою очередь, формирование признака осуществляется аналогично описанному выше случаю.

Алгоритм функционирования представленной схемы контроля представлен на рисунке 4.

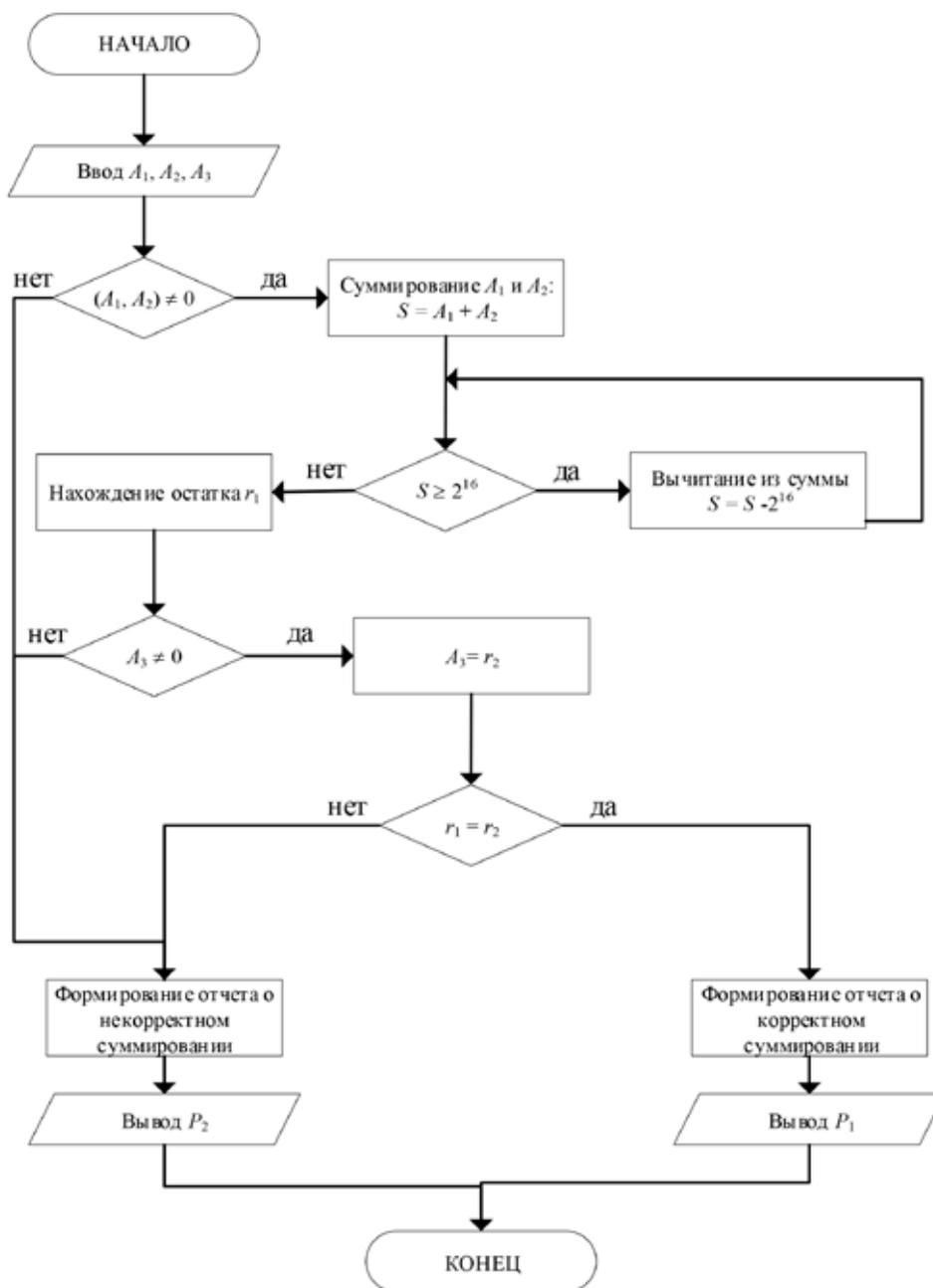


Рис. 4. Схема выполнения контроля операции сложения по модулю 2^{16}

Условно работу представленного алгоритма также можно разбить на несколько аналогичных этапов, представленных ранее. Отличие в данном случае заключается в отсутствии процедуры нахождения остатков при реализации отображений (3) и (4).

Следующим основным преобразованием, участвующим в алгоритмах КПИ, является операция сдвига. На рисунке 5 представлена обобщенная схема операции циклического сдвига.

Работу данной схемы можно описать следующим образом. Результатом сдвига числа A_1 есть число A_2 . Для блоков B_1 и B_2 осуществляется разделение битовой последовательности двоичного представления числа A_1 на два блока $A_1 = \{A_1^1, A_2^1\}$, которые в ходе выполнения операции будут переставлены относительно старшего и младшего разрядов (рис. 6).

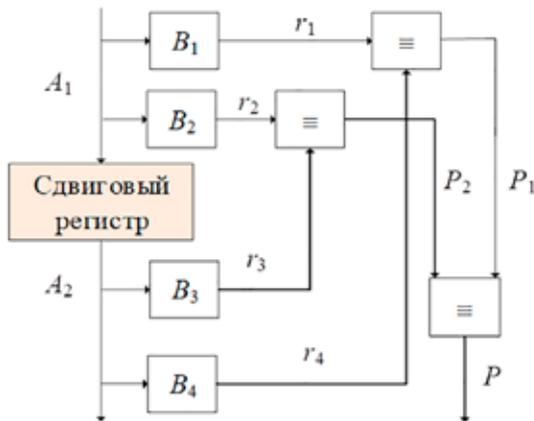


Рис. 5. Схема выполнения контроля операции сдвига

Поясним отдельно данное утверждение: в блоке B_1 преобразуется в остаток r_1 та часть двоичной последовательности (A_1^1), которая в дальнейшем будет участвовать в операции сдвига и в полном объеме бу-

дет смещена циклически в сторону младшего разряда; в свою очередь в блоке B_2 преобразуется в остаток r_2 та часть двоичной последовательности (A_2^1), которая будет смещена в сторону старшего разряда и будет переписана на месте двоичной последовательности числа A_1^1 . Далее в блоке сдвигового регистра происходит выполнение операции сдвига, в результате которой появляется число A_2 . Поскольку сдвиг является циклическим, смещенные биты не исчезают, а заполняют освободившиеся разряды с противоположной стороны битовой последовательности.

Исходя из этого, разделив битовую последовательность двоичного представления числа A_2 на A_1^2 и A_2^2 , возможно осуществить контроль правильности выполнения операции сдвига следующим образом. В блок B_3 (формирование остатка r_3) необходимо записать количество разрядов, равное количеству разрядов для блока B_2 . Аналогично заполняется блок B_4 (формирование остатка r_4). После выполнения операции сдвига, сравнив остатки r_1, r_4 и r_2, r_3 блоков B_1, B_2, B_3 и B_4 , можно сделать вывод о правильности выполнения циклического сдвига, тем самым осуществив контроль правильности выполнения операции. Необходимо отметить, что в блоках сравнения сопоставляются не целые значения чисел, а их остатки, что повышает скорость процесса сравнения сколь угодно больших блоков информации.

Алгоритм функционирования представленной схемы контроля представлен на рисунке 7. Работа представленного алгоритма может быть разбита на несколько этапов.

1. На начальном этапе осуществляются получение чисел A_1 и A_2 , участвующих в операции сдвига, а также разбиение числа A_1 на два блока A_1^1 и A_2^1 битовой последовательности. Также на данном этапе происходит нахождение остатков r_1 и r_2 в блоках B_1 и B_2 соответственно.

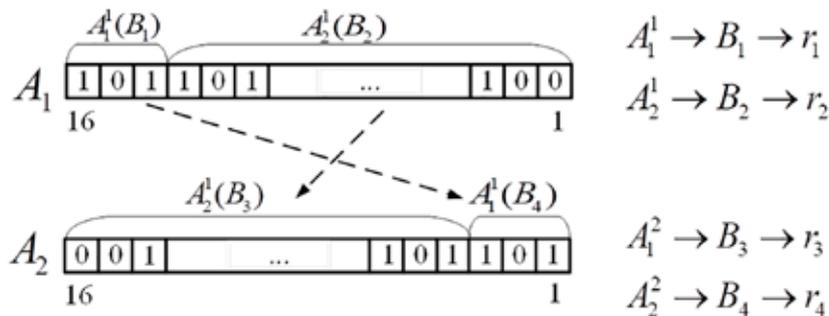


Рис. 6. Сдвиг в сторону старшего разряда на 3

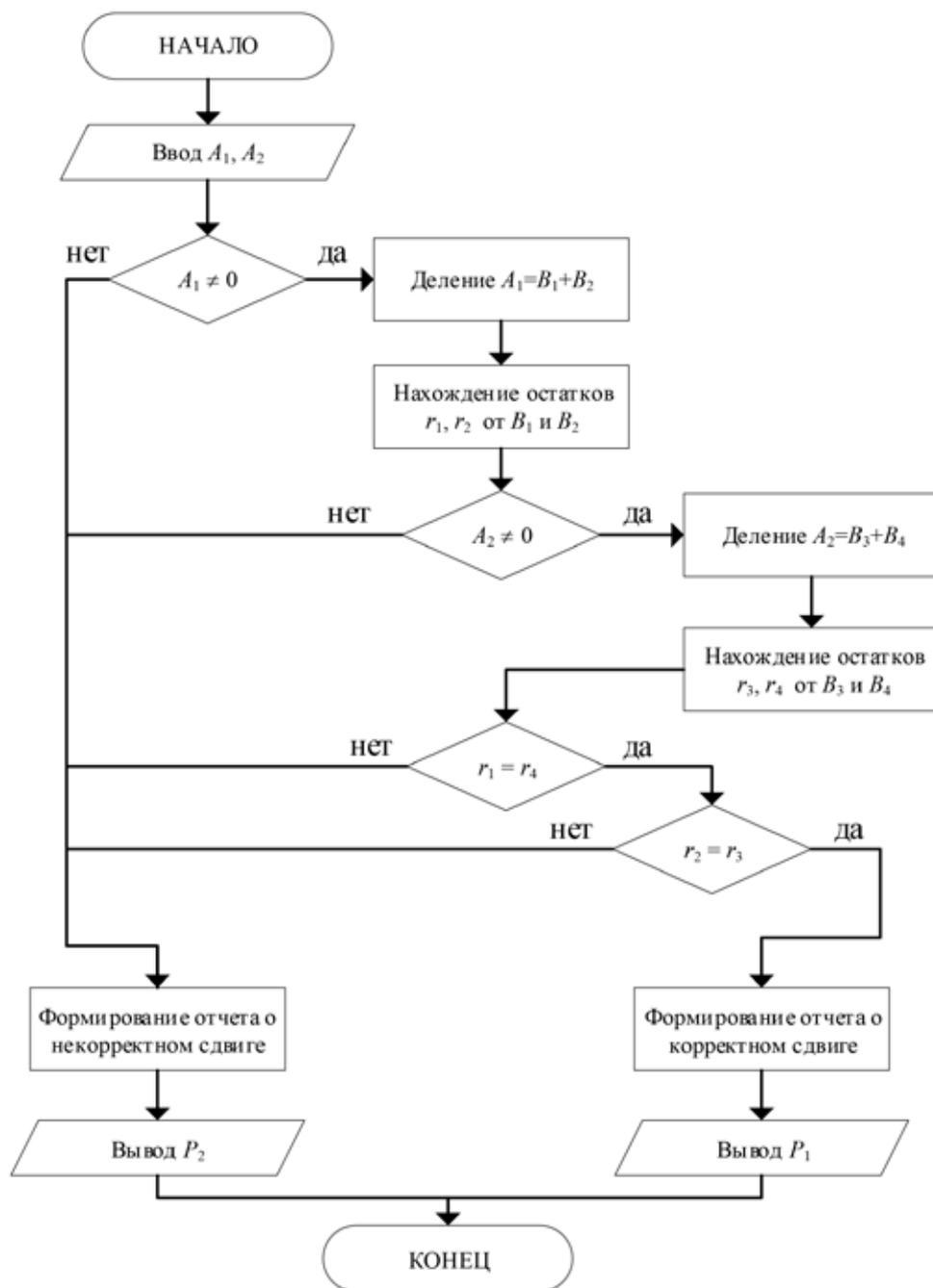


Рис. 7. Алгоритм выполнения контроля операции сдвига

2. Аналогично на следующем этапе осуществляются разбиение числа A_2 на два блока B_3 и B_4 битовой последовательности и нахождение остатков r_3 и r_4 от блоков B_3 и B_4 соответственно.

3. На заключительном этапе осуществляются сравнение остатков r_1 и r_4 от блоков битовой последовательности B_1 и B_4 , остатков r_2 и r_3 от блоков B_2 и B_3 , а также формирование отчета о корректном

или некорректном сдвиге (вывод P_1 и P_2 соответственно).

Представим данные этапы в виде совокупности отображений:

$$\alpha^* : A_1 \rightarrow \sum_{i=1}^2 A_i^1, \quad (9)$$

где α^* – оператор, характеризующий процедуру разбиения числа A_1 на битовые последовательности A_1^1 и A_2^1 соответственно;

$$\beta_1 : A_1^1 \rightarrow r_1, \quad (10)$$

где β_1 – оператор, характеризующий процедуру получения остатка r_1 от битовой последовательности A_1^1 в блоке B_1 ;

$$\beta_2 : A_2^1 \rightarrow r_2, \quad (11)$$

где β_2 – оператор, характеризующий процедуру получения остатка r_2 от битовой последовательности A_2^1 в блоке B_2 .

Необходимо отметить, что отображения β_3 и β_4 имеют схожую интерпретацию для остатков r_3 и r_4 .

$$\beta_3 : A_1^2 \rightarrow r_3, \quad (12)$$

где β_3 – оператор, характеризующий процедуру получения остатка r_3 от битовой последовательности A_1^2 в блоке B_3 ;

$$\beta_4 : A_2^2 \rightarrow r_4, \quad (13)$$

где β_4 – оператор, характеризующий процедуру получения остатка r_4 от битовой последовательности A_2^2 в блоке B_4 ;

$$\gamma : r_i \vee r_j \rightarrow p, p = \{0, 1\}, i = \overline{1, 2}, j = \overline{3, 4}, \quad (14)$$

где γ – оператор, формализующий процедуру сравнения r_1, r_4 и r_2, r_3 и принятие решения о корректном или некорректном сдвиге.

Последним рассматриваемым преобразованием является операция подстанов-

ки. Она заключается в том, что входная последовательность из n -ого количества бит разделяется на два блока с указанным количеством бит, эти блоки определяют адрес ячейки из таблицы замен, в которой записано значение, которое и будет являться результатом подстановки. Но с целью повышения криптостойкости в данном преобразовании применяется также и операция циклического сдвига выходной последовательности. Обобщенная схема контроля правильности выполнения данного преобразования представлена на рисунке 8.

Описание функционирования данной схемы выглядит следующим образом. Поступающее число A_1 , проходя через блок подстановки N , преобразуется в число A_2 , которое и является результатом выполнения операции подстановки. Число A_1 разбивается на два числа A_1^1 и A_1^2 , после чего они параллельно поступают в блок подстановки N и блок n . Блок n является таблицей остатков таблицы замены (блока N). Тем самым результат, полученный в ходе выполнения операции подстановки на выходе из блока n , будет представлен в остаточных классах. После этого над полученным результатом проводится операция циклического сдвига со сменой местами блоков двоичной записи результата.

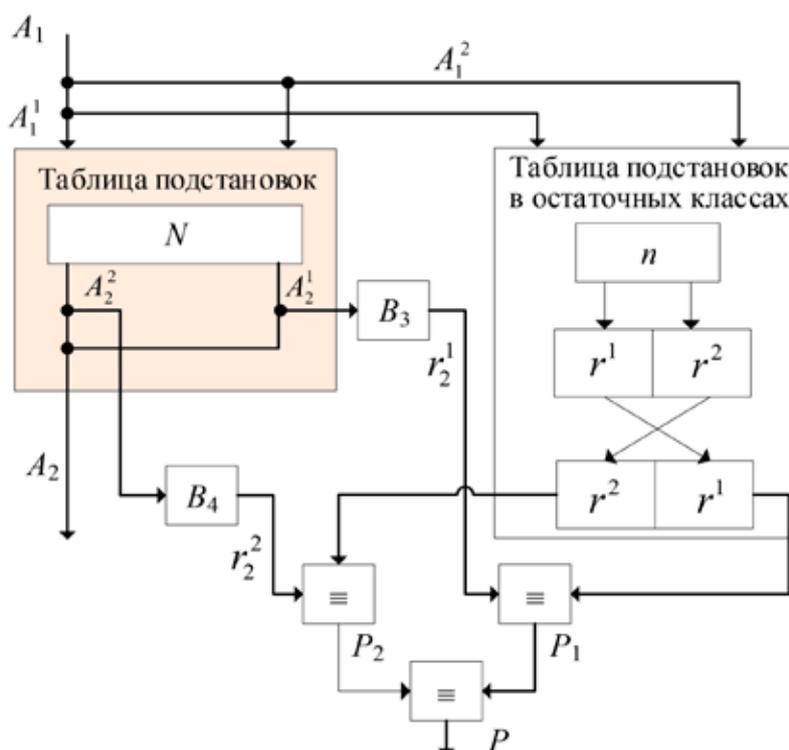


Рис. 8. Схема выполнения контроля операции подстановки

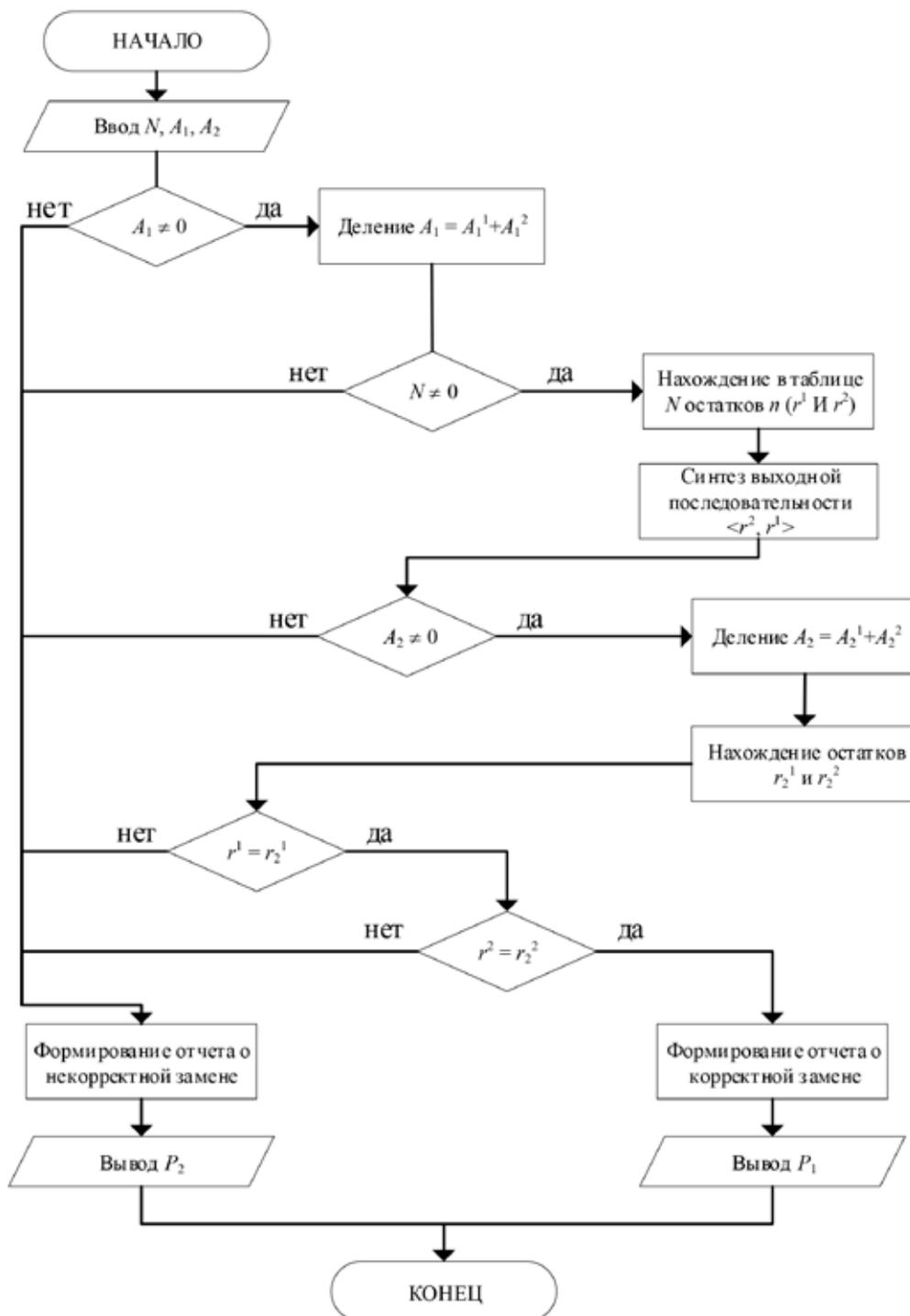


Рис. 9. Алгоритм выполнения контроля операции подстановки

Результатом прохождения чисел A_1^1 и A_1^2 через таблицу подстановок в остаточных классах n являются числа (остатки) r^1 и r^2 соответственно. После операции подстановки над блоками r^1 и r^2 выполняется операция циклического сдвига со сменой их местами. С целью контроля правильно-

сти выполнения преобразования подстановки число A_2 делят на блоки A_2^1 и A_2^2 . Далее в блоках B_3 и B_4 вычисляются числа (остатки) r_2^1, r_2^2 битовых последовательностей A_2^1 и A_2^2 , записанных в соответствующие блоки. Сравнение остатков r^1 и r^2, r_2^1 и r_2^2 в блоках сравнения (\equiv) с формированием

результатов P_1 и P_2 и P является результатом контроля правильности выполнения операции подстановки. В случае когда в результате сравнения получается, что $r^1 = r_2^1$ и $r^2 = r_2^2$, формируется отчет о правильности выполнения подстановки. В случае когда одно из равенств не выполняется либо не выполняются оба неравенства ($r^1 \neq r_2^1$, $r^2 \neq r_2^2$), следует вывод о неправильном выполнении контролируемого преобразования.

Алгоритм функционирования данной схемы контроля представлен на рисунке 9.

Работа представленного алгоритма может быть разбита на несколько этапов.

1. На первом этапе осуществляется ввод чисел A_1 , A_2 и таблица замен N . Затем A_1 делится на A_1^1 и A_1^2 . После этого находится таблица остатков n из таблицы замен N и находится выходная последовательность, представленная остатками r^1 и r^2 .

2. Второй этап начинается с разделения числа A_2 на блоки A_2^1 и A_2^2 таким образом, чтобы количество бит в блоках было равно количеству бит в блоках A_1^1 и A_1^2 соответственно. Далее из блоков A_2^1 и A_2^2 формируются остатки r_2^1 и r_2^2 .

3. На заключительном этапе остатки r^1 и r^2 сравниваются с r_2^1 и r_2^2 , после чего, если равенство выполняется для обоих выражений, формируется отчет о правильности выполнения операции P_1 , в противном случае формируется отчет о некорректном выполнении операции подстановки P_2 .

С формальной точки зрения данные этапы могут быть представлены совокупностью отображений:

$$\alpha_1^* : A_1 \rightarrow \sum_{i=1}^2 A_1^i, \quad (15)$$

где α_1^* – оператор, характеризующий процедуру разбиения числа A_1 на битовые последовательности A_1^1 и A_1^2 соответственно;

$$\beta_1 : N \rightarrow n, \quad (16)$$

где β_1 – оператор, характеризующий процедуру получения таблицы остатков n от таблицы замен N ;

$$\beta_2 : A_1^1 \rightarrow r^1, \quad (17)$$

где β_2 – оператор, характеризующий процедуру получения значения r^1 из таблицы остатков n от входной битовой последовательности A_1^1 ;

$$\beta_3 : A_1^2 \rightarrow r^2, \quad (18)$$

где β_3 – оператор, характеризующий процедуру получения значения r^2 из таблицы

остатков n от входной битовой последовательности A_1^2 ;

$$\alpha_2^* : A_2 \rightarrow \sum_{i=1}^2 A_2^i, \quad (19)$$

где α_2^* – оператор, характеризующий процедуру разбиения числа A_2 на битовые последовательности A_2^1 и A_2^2 соответственно;

$$\beta_4 : A_2^1 \rightarrow r_2^1, \quad (20)$$

где β_4 – оператор, характеризующий процедуру получения значения r_2^1 из битовой последовательности A_2^1 ;

$$\beta_5 : A_2^2 \rightarrow r_2^2, \quad (21)$$

где β_5 – оператор, характеризующий процедуру получения значения r_2^2 и битовой последовательности A_2^2 ;

$$\gamma : r^i \vee r^j \rightarrow p,$$

$$p = \{0, 1\}, i = \overline{1, 2}, j = \overline{3, 4}, \quad (22)$$

где γ – оператор, формализующий процедуру сравнения r^1 и r_2^1 , r^2 и r_2^2 , а также принятие решения о корректном или некорректном сдвиге.

Заключение

В ходе исследований были разработаны алгоритмы контроля основных операций алгоритма криптографического преобразования информации на основе модулярной арифметики и в соответствии с предложенными схемами контроля. Указанные алгоритмы лежат в основе формирования общего алгоритма технического диагностирования программируемых логических интегральных схем на автономных технических средствах.

Список литературы

1. Рыжов И.А., Харитонов А.С., Столяров А.В., Семенов Д.Б. Оценка качества радиоканала передачи телеметрической информации // Инновационная наука: сборник статей по материалам XXII научно-практической конференции. 2019. Т. 4 (22). С. 65-73.
2. Cheng L., Huberman B.A. Auction-based Operation in LEO Satellite Systems for High-Efficiency Communications // IEEE Wireless Communications, 2020. Vol. 27, №. 2. P. 2-8.
3. Лоскутов А.И., Ряхова Е.А., Горбулин В.И. Концептуальная модель технического диагностирования бортовой аппаратуры автономных космических аппаратов на основе оптимальной реконфигурации в условиях априорной неопределенности появления неисправностей // Информационно-измерительные и управляющие системы. 2020. № 3. С. 43-55.
4. Охтилев М.Ю., Соколов Б.В., Юсупов Р.М. Теоретические и технологические основы концепции проактивного

мониторинга и управления сложными объектами // Известия ЮФУ. Технические науки. 2015. № 1(162). С. 162–174.

5. Лоскутов А.И., Клыков В.А. Идентификация и техническое диагностирование бортовой аппаратуры автономных космических аппаратов на основе биективного преобразования множества диагностических признаков // Контроль диагностика. 2016. № 4. С. 57-63.

6. Семенюк Д.Б., Лоскутов А.И., Бардаев Э.А., Клыков В.А. Системное моделирование при разработке математического обеспечения автоматизированных комплексов на основе полимодельного подхода с иерархическим прин-

ципом // Авиакосмическое приборостроение. М.: Научтехлитиздат, 2019. № 4. С. 33-44.

7. Столяров А.В., Никулин В.А., Лоскутов А.И., Клыков В.А., Ряхова Е.А. Методика построения математической модели процесса функционирования беспилотного авиационного комплекса с целью решения задачи технического диагностирования // Надежность и качество сложных систем. 2020. С. 45-67.

8. ГОСТ Р 34.13 – 2015. Информационная технология. Криптографическая защита информации. Режимы работы блочных шифров. М.: Стандартинформ, 2015.

УДК 621.396.6

ОПТИМИЗАЦИЯ КОНСТРУКЦИИ ФИЛЬТРА СИНТЕЗАТОРА ЧАСТОТ РАДИОКОМПЛЕКСА СВЯЗИ КОСМИЧЕСКОГО АППАРАТА**¹Шмелькова А.А., ¹Андреев В.В., ²Гурин В.М., ³Драч В.Е.**¹ФГБОУ ВО «Московский государственный технический университет имени Н.Э. Баумана, Калужский филиал, Калуга, e-mail: nastay31051999@gmail.com;²ООО «Фокон», Калуга, e-mail: vitally.gurin918@yandex.ru;³ФГБОУ ВО «Сочинский государственный университет», Сочи, e-mail: rpd-admin@inbox.ru

Четкое соответствие теоретических и расчетных параметров приборов электронной техники реальным параметрам изготовленного прибора – важнейший вопрос для современного инженера-конструктора электронной аппаратуры, особенно работающего с космической техникой, где цена ошибки возрастает до невероятных высот. Основные разработки в космической сфере являются продуктом работ многих лет. Электроника развивается в быстром темпе, и удачные схемотехнические решения требуют адаптации с учетом новых стандартов приборостроения. При разработке фильтров, если учитывать паразитные связи между элементами, параметры сконструированного прибора часто не соответствуют теоретически ожидаемым. В настоящей статье проведен анализ и оптимизация конструкции фильтров для синтезатора частот, работающего в блоке приема и передачи радиочастотного сигнала космического аппарата. Целью работы является выявление расхождений расчетных параметров и параметров изготовленного прибора, а также определение причин данных расхождений и разработка рекомендаций по их минимизации. Проведено моделирование с учетом паразитных параметров в симуляторе электронных цепей. Составлены рекомендации по доработке и улучшению конструкции фильтров. Детально исследовано и промоделировано влияние элементов механического крепления на электромагнитные характеристики электрорадиоэлементов.

Ключевые слова: космический аппарат, фильтр, подавление помех, петлевой фильтр, синтезатор частот**FILTER DESIGN OPTIMIZATION OF THE FREQUENCY SYNTHESIZER OF THE SPACECRAFT COMMUNICATION RADIO COMPLEX****¹Shmelkova A.A., ¹Andreev V.V., ²Gurin V.M., ³Drach V.E.**¹Bauman Moscow State Technical University, Kaluga branch, Kaluga, e-mail: nastay31051999@gmail.com;²LLC «Fokon», Kaluga, e-mail: vitally.gurin918@yandex.ru;³Sochi State University, Sochi, e-mail: rpd-admin@inbox.ru

A clear correspondence of the theoretical and design parameters of electronic devices to the real parameters of the manufactured device is the most important issue for a modern electronic equipment design engineer, especially working with space technology, where the cost of error increases to incredible heights. The main developments in the space sector are the product of many years of developments. Electronics is developing at a fast pace and successful circuit solutions require adaptation to new instrumentation standards. When developing filters, if many parasitic connections are not taken into account, the parameters of the constructed device often do not correspond to theoretically predicted ones. This article analyzes and optimizes the design of filters for a frequency synthesizer operating in the radio frequency signal receiving and transmitting unit of a spacecraft. This article analyzes and optimizes the design of filters for a frequency synthesizer operating in the receiving and transmitting unit of the radio frequency signal of the spacecraft. The purpose of the work is to identify discrepancies between the design parameters and the parameters of the manufactured device, as well as to determine the causes of these discrepancies and develop recommendations for their minimization. The simulation is carried out taking into account parasitic parameters in the simulator of electronic circuits. Recommendations on the refinement and the improvement of the filter design have been compiled. The influence of mechanical fastening elements on the electromagnetic characteristics of electric radio elements has been studied in detail and modeled.

Keywords: spacecraft, filter, interference suppression, loop filter, frequency synthesizer

Несмотря на то, что исследования естественного спутника Земли – Луны – начались достаточно давно [1, 2], в настоящее время исследователи испытывают нехватку данных об этом объекте [3]. Поэтому разработка исследовательских аппаратов является актуальной задачей на данный момент. Для космических аппаратов, создаваемых по проекту исследования Луны [4, 5], разрабатывается радиокомплекс связи орбитального аппарата с посадочным аппаратом, расположенным на поверхности. В составе одного из функ-

циональных блоков комплекса, отвечающего за прием и передачу радиочастотного сигнала, применяется синтезатор частот, требовательный к питающему напряжению.

При разработке электронных блоков в космических аппаратах, в частности, отвечающих за прием и передачу радиочастотного сигнала [6], большое внимание уделяется выбору высоконадежной элементной базы, способной работать в различных критических условиях, в том числе и при воздействии радиационных излучений [7, 8].

Из-за разброса параметров компонентов фильтра или их паразитных параметров, а также из-за наличия собственной емкости и индуктивности монтажа зачастую параметры разработанного и собранного изделия не в полной мере соответствуют ожидаемым. Актуальной является задача нахождения причины пониженного коэффициента затухания помехоподавляющего фильтра, примененного в модуле связи космического летательного аппарата, при условии, что проверка по методике, утвержденной в технических условиях, не выявляет отклонений. Расхождение параметров можно отследить, применив прогрессивные комбинированные методики для исследования изготовленных на предприятии фильтров. Деграцией элементов [9–11], обусловленной излучением в условиях космоса, пренебрегаем.

Материалы и методы исследования

В качестве объекта исследования выступает фильтр синтезатора частот радиокомплекса связи космического аппарата (далее – фильтр). В данной работе будут определены причины расхождения параметров фильтра и выдвинуты предложения по параметрической оптимизации прибора, для чего потребуется комбинировать лабораторные измерения реальных параметров и рассмотрение результатов симуляции компьютерной модели. Компьютерное моделирование выполнено в актуальной на данный момент версии системы автоматизированного проектирования QUCS [12], которая достоверно моделирует электрические схемы [13], в том числе фильтры [14].

Схема электрическая принципиальная рассматриваемого фильтра представлена на рис. 1, такое решение является классическим для подавления симметричных и несимметричных помех. Изменение структу-

ры в данном случае допускается в крайних случаях, поэтому усилия следует направить на параметрическую оптимизацию. Элементы Э на рис. 1 – это стандартные проходные керамические фильтры модели Б14, которые предназначены для подавления помех в верхней части частотного диапазона.

Требуемый коэффициент затухания в пределах рабочего диапазона должен составить не менее 60 дБ для частот 0,05–400 МГц. Промоделируем фильтр с идеальными параметрами, чтобы иметь эталонную амплитудно-частотную характеристику в качестве референса. Амплитудно-частотная характеристика одного канала фильтра представлена на рис. 2 (для второго канала результат будет идентичным). Здесь не учитываются ни взаимное влияние элементов, ни паразитные параметры компонентов. Следует отметить, что смоделированная амплитудно-частотная характеристика демонстрирует лишь работоспособность фильтра в общем случае и показывает характер отклика, но не дает количественной оценки его затухания. Оценить значения коэффициента затухания можно лишь на низких частотах, где еще не сказываются нелинейные параметры компонентов (это частоты до ~75 кГц в нашем случае). Полюс фильтра располагается вблизи 15 кГц, резкий скачок вблизи 3 МГц обусловлен резонансом среднего звена. Чтобы оценить параметры реального устройства, требуется дополнить компьютерную модель паразитными параметрами, которые являются неотъемлемой частью ВЧ-прибора в реальном мире.

Наибольшие трудности в лабораторных измерениях возникли при определении значений параметров, которые могут быть измерены лишь косвенными методами: емкость дросселя и эквивалентное последовательное сопротивление конденсатора (англ. Equivalent Serial Resistance – ESR).

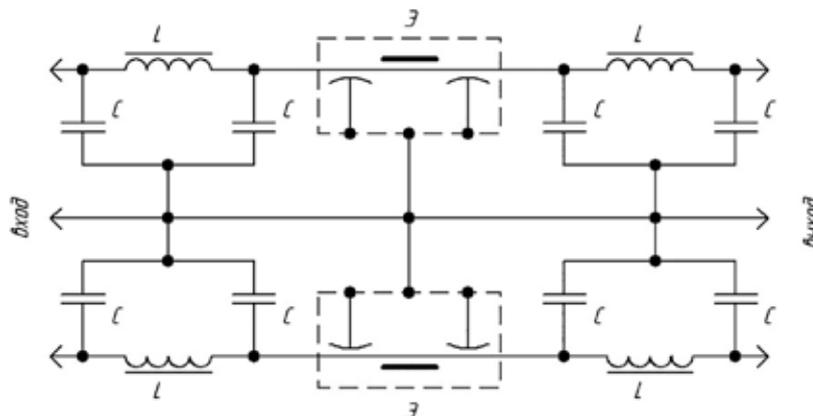


Рис. 1. Схема электрическая принципиальная помехоподавляющего фильтра

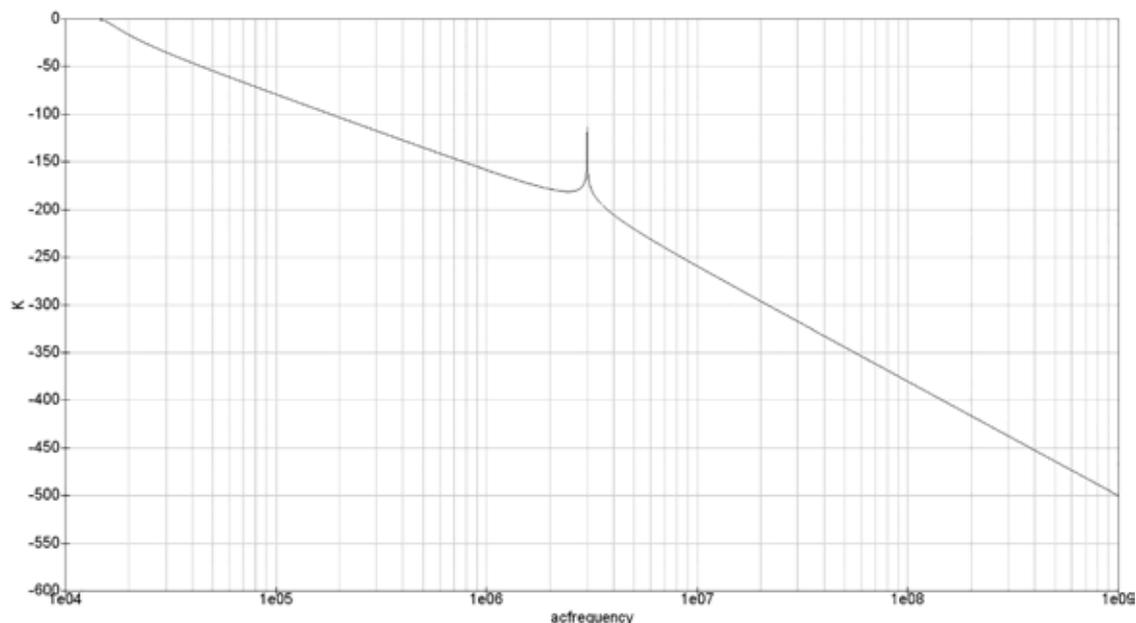


Рис. 2. Амплитудно-частотная характеристика идеального фильтра

Эквивалентное последовательное сопротивление конденсатора, как правило, не превышает доли Ом для высококачественных конденсаторов, но может достигать единиц Ом для особых партий или неудачных образцов. Для электролитических конденсаторов оно играет особенно важную роль. В лабораторных условиях эквивалентное последовательное сопротивление обычно измеряется путем подачи прямоугольного меандра амплитудой около 10 Вольт и частотой около 50 кГц на тестовый конденсатор, последовательно соединенный с низкоомным резистором, на котором измеряется падающее напряжение переменного тока.

Измерение паразитной емкости дросселя можно провести несколькими способами. Например, путем определения добротности (Q) колебательного контура, образованного индуктивностью и ее паразитной емкостью.

В рассматриваемом случае в контрольной выборке из партии компонентов Э средние значения измеренных параметров составили: 17 нГн для индуктивности фильтра и 1,67 пФ для паразитной емкости. Далее была измерена емкость у 20 дросселей, среднее значение составило 82,6 пФ.

Компьютерная модель фильтра, с учетом вышеизложенного, усложняется и принимает вид, приведенный на рис. 3. В этой модели учтены паразитные эквивалентное последовательное сопротивление конденсаторов, емкости и собственное сопротивление низкочастотных дросселей и керамического фильтра. Моделирование проведено для диапазона 10 кГц – 1 ГГц при входном и выходном сопротивлении равном 50 Ом.

На низкой частоте значения коэффициента затухания промоделированы весьма достоверно, поэтому уже на данном этапе исследования можно отметить, что требования технических условий не выполняются.

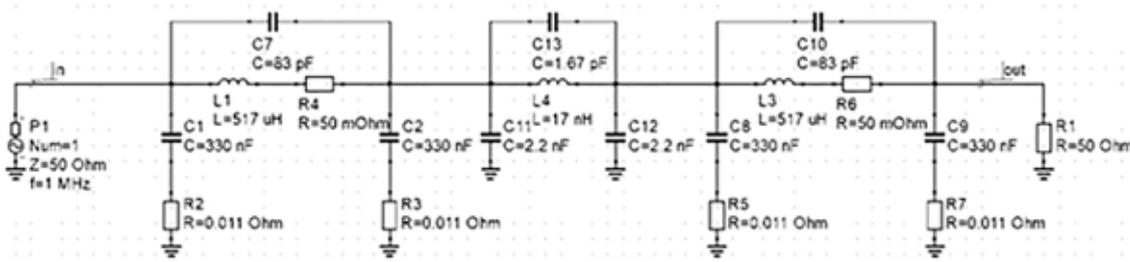


Рис. 3. Схема электрическая принципиальная дополненной компьютерной модели

На частотах порядка 50 кГц можно улучшить затухание двумя путями: увеличив индуктивность дросселей или повысив емкость конденсаторов. Физические габариты корпуса жестко фиксированы для космического аппарата, поэтому для конкретного прибора рационально увеличение индуктивности дросселей (добавление определенного количества витков на магнитопровод), нежели наращивание емкости за счет установки массивных электролитов. Принимаем новое значение индуктивности в 1 мГн, что приводит к новому значению коэффициента затухания в -66 дБ, которое входит в допустимый диапазон. На высоких частотах найти причины несоответствия коэффициента затухания требуемому можно, построив амплитудно-частотную характеристику фильтра в режиме холостого хода и короткого замыкания, чтобы оценить влияние импеданса нагрузки. Проведем моделирование в диапазоне 100 Гц – 1 ГГц, изменяя значение сопротивления с помощью развертки параметра от 50 кОм (для режима холостого хода) и 0,5 Ом (для режима короткого замыкания).

При поиске возможных решений проблемы несоответствия параметров была выдвинута гипотеза о ключевом влиянии емкости керамических фильтров Б14 на коэффициент затухания. В результате была произведена отбраковка по емкости всех имеющихся в наличии фильтров, а в изделии были установлены фильтры с большими значениями емкости.

Оценим, как сильно на частотную характеристику изделия влияет емкость керамического фильтра в пределах допуска. Параметры моделирования остаются аналогичными, как и при исследовании режимов короткого замыкания и холостого хода. За счет изменения емкости керамического фильтра в диапазоне 1,8–2,8 нФ, удалось построить 6 амплитудно-частотных характеристик, почти полностью накладывающихся друг на друга. Таким образом, можно сделать заключение, что изменение емкости фильтра Б14 в пределах допуска не оказывает какого-либо влияния на частотную характеристику прибора.

Результаты исследования и их обсуждение

Излучающие свойства элементов конструкции начинают проявляться на частотах выше 200 МГц. Например, металлический винт, находящийся вблизи дросселя, начинает играть роль штыревой антенны: часть энергии отбирается и излучается в окружающее пространство. Если винт проходит через несколько экранированных отсеков устройства, то он превращается

в источник помех, так как становится проводником электромагнитного поля. В таком случае экранирование становится малоэффективным. Более того, крепежные элементы конструкции, которые можно представить как короткозамкнутые витки, будут мешать нормальной работе устройства на высокой частоте, что обусловлено переизлучением энергии.

Фильтр, рассматриваемый в данной работе, состоит из двух отсеков (экранированных стенками и перегородкой), в каждом из которых находится по два низкочастотных П-образных звена: половины верхнего канала и нижнего по схеме (рис. 1). Отсеки соединяются через керамический фильтр Б14. Дроссели каждого канала механически сопрягаются через общий винт, проходящий через оба отсека. Металлические клипсы применены для механического крепления низкочастотных конденсаторов на перегородке.

Общие крепежные винты для дросселей, очевидно, можно считать элементами, привносящими паразитные параметры в рассматриваемую конструкцию. Играя роль антенны, винт дополнительно выступает как общий магнитопровод для двух дросселей. Таким образом, для помех появляется обходной путь вокруг керамического фильтра, что приводит к уменьшению коэффициента затухания. Помимо этого, металлические клипсы крепления конденсаторов также начинают привносить паразитные параметры, поскольку они являются короткозамкнутыми витками.

Для оценки влияния элементов конструкции на частотные характеристики усложним модель, добавив элементы, как показано на рис. 4. Для создания общего магнитного поля между двумя катушками последовательно с ними включен трансформатор Tr1 с практически идеальными характеристиками: с единичным коэффициентом трансформации и малой индуктивностью обмоток. Трансформаторы Tr2-Tr5 с параметрами, аналогичными Tr1, включены последовательно с низкочастотными конденсаторами. Их первичная обмотка представляет индуктивность обкладок конденсатора, а вторичная обмотка – короткозамкнутый виток крепежной клипсы.

В итоге промоделирована амплитудная характеристика, представленная на рис. 5.

При таком подходе становится видно, что в диапазоне от 5 до 100 МГц коэффициент затухания резко спадает и становится намного меньше требуемого в -60 дБ. В других областях частотной характеристики затухание также ощутимо ослабло, хотя и осталось в пределах допуска.

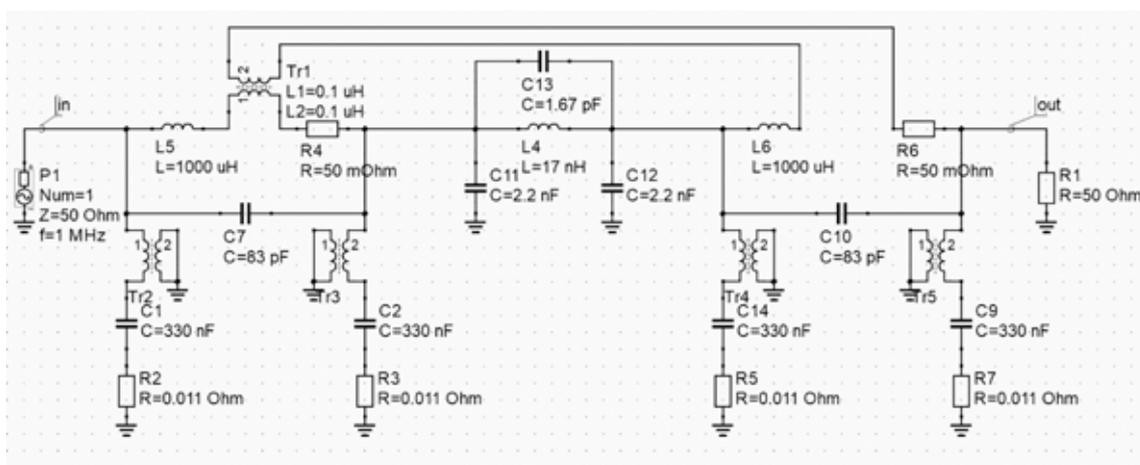


Рис. 4. Схема электрическая принципиальная фильтра в финальном варианте

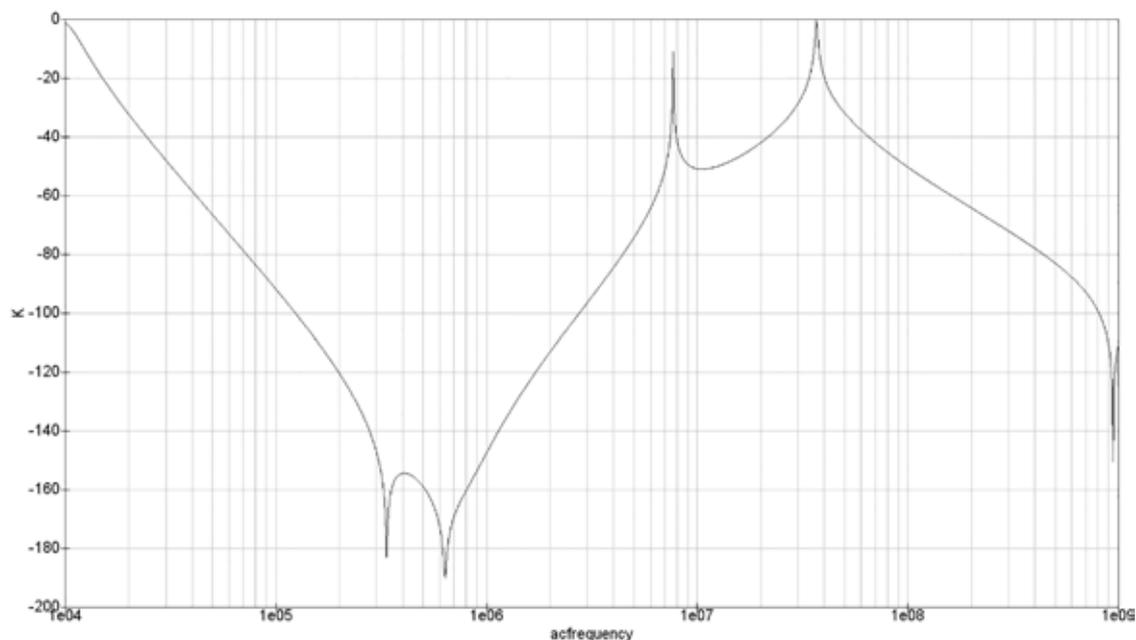


Рис. 5. Результирующая амплитудно-частотная характеристика компьютерной модели

Заключение

В результате исследования было установлено, что рассматриваемый фильтр по своим параметрам не обеспечивает необходимого затухания в высокочастотной и низкочастотной областях диапазона. Требуется увеличить индуктивность низкочастотных дросселей для увеличения коэффициента затухания в нижней области диапазона. Параметры керамических фильтров и емкостей не оказывают существенного влияния на работу прибора. Заметно ухудшают характеристики фильтра, как оказалось, крепления низкочастотных дросселей, (обеспечивающих паразитную емкостную связь между отсеками) и крепления

конденсаторов (представляющие собой короткозамкнутый виток). Требуется изменить способ крепления компонентов для оптимизации электрофизических параметров. Можно предложить для достижения этой цели заменить материал крепления дросселя (с металла на полиамид). В качестве материала крепления дросселей необходимо выбирать диэлектрик.

Список литературы

1. Черкасова Л.И. История первых исследований механических свойств грунтов Луны // Основания, фундаменты и механика грунтов. 2022. № 3. С. 15–18.
2. Черкасова Л.И. Исследования грунтов Луны. История и перспективы // Вестник МГСУ. 2011. № 5. С. 301.

3. Попандопуло Н.А., Александрова А.Г., Томилова И.В., Авдюшев В.А., Бордовицына Т.В. Численное моделирование динамики искусственных спутников Луны // *Астрономический вестник. Исследования Солнечной системы*. 2022. Т. 56, № 4. С. 266–284.
4. Рубцова А.С. Исследование системы управления посадкой космического летательного аппарата «Луна-Глоб» на участке прецизионного торможения // *Автоматизация. Современные технологии*. 2022. Т. 76, № 4. С. 179–184.
5. Дудоров Е.А. Робототехнические системы космического назначения // *Космическая техника и технологии*. 2022. № 3 (38). С. 66–81.
6. Коровин А.К. Математическая модель системы функционирования космических аппаратов связи // *Стратегическая стабильность*. 2021. № 1 (94). С. 2–6.
7. Абдурахимов А.А., Скворцов Д.В. Моделирование деградации крупногабаритных уязвимых элементов космических аппаратов в условиях воздействия факторов космического пространства // *Труды Военно-космической академии имени А.Ф. Можайского*. 2016. № 650. С. 132–137.
8. Esqueda I.S., Varnaby H.J., King M.P. Compact modeling of total ionizing dose and aging effects in MOS technologies // *Transactions on Nuclear Science*. 2015. Vol. 62, Is. 4. P. 1501–1515.
9. Драч В.Е., Смирнова О.М., Чухраев И.В. Генерация заряда в транзисторах с наноразмерным диэлектриком // *Вопросы радиоэлектроники*. 2012. Т. 1, № 3. С. 115–122.
10. Андреев В.В., Бондаренко Г.Г., Дегтярев В.Т., Столяров А.А., Драч В.Е. Моделирование процессов сильно-полевой инжекционной модификации и деградации МДП-структур // *Перспективные материалы*. 2004. № 2. С. 20–27.
11. Andreev D.V., Bondarenko G.G., Andreev V.V., Maslovsky V.M., Stolyarov A.A. Modification of MIS Devices by Radio-Frequency Plasma Treatment // *Acta Phys. Pol. A*. 2019. Vol. 136, Is. 2. P. 263–266.
12. Кечиев Л.Н., Крючков Н.М., Кузнецов В.В. Симулятор QUCS: основные возможности // *Инновации на основе информационных и коммуникационных технологий*. 2015. Т. 1. С. 398–400.
13. Драч В.Е., Корчинова А.Е. Повышение уровня стабилизации драйвера светодиодов // *Вестник инженерной школы Дальневосточного федерального университета*. 2016. № 1 (26). С. 10–16.
14. Дубинина М.С., Рыжов С.В., Крючков Н.М., Кузнецов В.В. Инструмент для синтеза фильтра Лежандра и катушек индуктивности в САПР QUCS // *Технологии электромагнитной совместимости*. 2017. № 4 (63). С. 42–47.