**Санкт-Петербургский государственный электротехнический университет**

**“ЛЭТИ” им. В. И. Ульянова (Ленина)**

**(СПбГЭТУ “ЛЭТИ”)**

|  |
| --- |
|  |

**Направление подготовки:** 09.03.01 “Информатика и вычислительная техника”

**Профиль:** “Вычислительные машины, комплексы, системы и сети”

**Факультет компьютерных технологий и информатики**

**Кафедра вычислительной техники**

*К защите допустить:*

**Заведующий кафедрой**

д. т. н., профессор\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ М. С.Куприянов

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА**

**БАКАЛАВРА**

**Тема: “** **Реализация функции извлечения квадратного корня на**

**ПЛИС”**

Студент \_\_\_\_\_\_\_\_\_\_\_\_\_ В. А. Приемышев

Руководитель

к. т. н., доцент \_\_\_\_\_\_\_\_\_\_\_\_\_ О. И. Буренева

Консультант

д. э. н., профессор \_\_\_\_\_\_\_\_\_\_\_\_\_ Т. Д. Маслова

Консультант от кафедры

к. т. н., доцент \_\_\_\_\_\_\_\_\_\_\_\_\_ И. С. Зуев

Санкт-Петербург

2019 г.

**Санкт-Петербургский государственный электротехнический университет**

**“ЛЭТИ” им. В. И. Ульянова (Ленина)**

**(СПбГЭТУ “ЛЭТИ”)**

|  |  |
| --- | --- |
| Направление: 09.04.01 “Информатика и вычислительная техника”  Профиль: “Вычислительные машины,  комплексы, системы и сети”  Факультет компьютерных технологий  и информатики  Кафедра вычислительной техники | **УТВЕРЖДАЮ**  Заведующий кафедрой ВТ  д. т. н., профессор  (М. С. Куприянов)  “\_\_\_” \_\_\_\_\_\_\_\_\_\_ 201\_\_г. |

**ЗАДАНИЕ**

**на выпускную квалификационную работу**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Студент | **В. А. Приемышев** |  | Группа № | **5305** |

|  |  |
| --- | --- |
| **1. Тема** | **Реализация функции извлечения квадратного корня на** |
| **ПЛИС** | |
| *(утверждена приказом № \_\_\_\_\_ от \_\_\_\_\_\_\_\_\_\_\_\_\_)* | |

Место выполнения ВКР: Санкт-Петербургский государственный электротехнический университет «ЛЭТИ» им. В. И. Ульянова (Ленина)

**2. Объект и предмет исследования:** конвейерный матричный делитель и специализированный кремниевый компилятор.

**3. Цель**: изучение технологии проектирования топологии макроблоков интегральных схем в технологически инвариантной концепции.

**4. Исходные данные:** Алгоритм деления – без восстановления остатка. Средства проектирования – САПР TopDesign, Matching of Cells. Схема сумматора Hample.

**5. Содержание:** Изучение алгоритма деления и разработка структурно-топологического плана матричного делителя. Применение системы TopDesign (технологически инвариантное проектирования топологии). Изучение схемных решений сумматора. Структурно-топологическая оптимизация КМД. Применение системы технологически инвариантного иерархического проектирования топологий макроблоков Matching of Cells. Разработка топологической реализации ячеек. Разработка спецификации ячеек. Разработка спецификаций макроблока КМД. Разработка программы генерации спецификации, параметризованной по разрядности операндов. Генерация топологий КМД для разных разрядностей в разных проектных нормах.

**6. Технические требования:** технологическая инвариантность, чётная разрядность операндов.

**7. Дополнительные разделы:** составление бизнес-плана по коммерциализации результатов НИР магистранта.

**8. Результаты:** пояснительная записка, иллюстративный материал.

|  |  |  |
| --- | --- | --- |
| Дата выдачи задания |  | Дата представления ВКР к защите |
| « 1 » сентября 2017 г. |  | « 31 » мая 2019 г. |

|  |  |  |
| --- | --- | --- |
| Руководитель  к. т. н., доцент | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | О. И. Буренева |
| Студент | **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** | В. А. Приемышев |

**Санкт-Петербургский государственный электротехнический университет**

**“ЛЭТИ” им. В. И. Ульянова (Ленина)**

**(СПбГЭТУ “ЛЭТИ”)**

|  |
| --- |
|  |

|  |  |
| --- | --- |
| Направление: 09.04.01 “Информатика и вычислительная техника”  Профиль: “Вычислительные машины,  комплексы, системы и сети”  Факультет компьютерных технологий  и информатики  Кафедра вычислительной техники | **УТВЕРЖДАЮ**  Заведующий кафедрой ВТ  д. т. н., профессор  (М. С. Куприянов)  “\_\_\_” \_\_\_\_\_\_\_\_\_\_ 201\_\_г. |

**КАЛЕНДАРНЫЙ ПЛАН**

**выполнения выпускной квалификационной работы**

|  |  |
| --- | --- |
| Тема | **Реализация функции извлечения квадратного корня на** |
| **ПЛИС** | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Студент | **В. А. Приемышев** |  | Группа № | **5305** |

|  |  |  |
| --- | --- | --- |
| № этапа | Наименование работ | Срок  выполнения |
| 1 | Изучение алгоритмов деления, поиск и работа с литературой по теме | 01.09.2017 – 31.12.2017 |
| 2 | Структурно-топологическое проектирование КМД | 08.01.2018 – 11.03.2018 |
| 3 | Схемотехническое проектирование ячеек КМД | 12.03.2018 – 31.05.2018 |
| 4 | Разработка топологической реализации ячеек КМД | 03.09.2018 – 30.12.2018 |
| 5 | Изучение методов, средств сжатия и согласования ячеек КМД по габаритам и положению выводов | 07.01.2019 – 10.02.2019 |
| 6 | Разработка спецификации ячеек. Разработка спецификации макроблока КМД | 11.02.2019 – 05.04.2019 |
| 7 | Генерация топологий КМД для разных разрядностей в разных проектных нормах | 06.04.2019 – 05.05.2019 |
| 8 | Оформление пояснительной записки | 06.05.2019 – 19.05.2019 |
| 9 | Оформление иллюстративного материала | 20.05.2019 – 26.06.2019 |
| 10 | Предварительное рассмотрение работы | 27.05.2019 – 30.05.2019 |
| 11 | Представление работы к защите | 31.05.2019 |

|  |  |  |
| --- | --- | --- |
| Руководитель  к. т. н., доцент | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | О. И. Буренева |
| Студент | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | В. А. Приемышев |

**РЕФЕРАТ**

Пояснительная записка 00 стр., 00 рис., 00 табл., 00 ист., 00 прил.

Ключевые слова и словосочетания, не более пятнадцати, через запятую

Объектом исследования (разработки) являются указать объект исследования или разработки.

Цель работы – кратко (в 2-3 строки) указать цель работы.

Кратко (в 10-12) строк описать основное содержание работы, методы исследования (разработки), полученные результаты.

**ABSTRACT**

Briefly (10-15 lines) the content of graduating work is specified

## Перечень основных сокращений

АК – антифонная катушка

ДМ – динамический микрофон

ЗК – звуковая катушка

МЦ – магнитная цепь

ПН – полюсный наконечник

ПС подвижная система

СЗУ – система звукоусиления

ТЗ – техническое задание

ХН – характеристика направленности

ЧХЧ – частотная характеристика направленности

# Содержание

[Введение 7](#_Toc388282647)

[1 Ненаправленные катушечные динамические микрофоны 8](#_Toc388282648)

1.1 [Однонаправленный динамический микрофон 14](#_Toc388282649)

[1.2 Постановка задачи – Техническое задание 17](#_Toc388282652)

[2 Расчет подвижной системы 19](#_Toc388282653)

[2.1 Расчет звуковой катушки 19](#_Toc388282654)

[2.2 Расчет параметров диафрагмы 21](#_Toc388282655)

[2.3 Расчет массы и резонансной частоты подвижной системы 23](#_Toc388282656)

[3 Расчет магнитной системы 24](#_Toc388282658)

[3.1 Расчет магнита 25](#_Toc388282659)

[3.2 Расчет индукции в зазоре 26](#_Toc388282660)

[4 Расчет конструктивных и акустико-механических параметров капсюля 29](#_Toc388282661)

[4.1 Параметры подкупольного объема и зазора 29](#_Toc388282659)

[4.2 Параметры внутренней структуры капсюля 30](#_Toc388282660)

[4.3 Параметры второго входа 31](#_Toc388282659)

[4.4 Расчет чувствительности, уровня собственного шума, предельного уровня звукового давления 34](#_Toc388282660)

[4.5 Расчет антифонной катушки 36](#_Toc388282659)

[5 Расчет частотно-пространственной характеристики чувствительности 39](#_Toc388282662)

[6 Организация рабочего места инженера 45](#_Toc388282662)

[Заключение 53](#_Toc388282662)

[Список литературы 54](#_Toc388282663)

# Введение

Деление и квадратный корень всегда охватывали основные роли в компьютерной арифметике. Их важность стала более актуальной с недавнего времени.

Квадратный корень является одной из наиболее полезных и жизненно важных операций в приложениях компьютерной графики и научных вычислений, таких как алгоритмы цифровой обработки сигналов (DSP), математический сопроцессор, обработка данных, управление и даже мультимедийные приложения.

Это классическая проблема в вычислительной теории чисел, которая очень часто встречается, что является трудной задачей для получения точного результата. Было изучено, разработано и внедрено множество алгоритмов квадратного корня, таких как грубая оценка, метод Вавилона, цифра за цифрой, алгоритм разложения в ряд Тейлора, метод Ньютона-Рафсона и т.д. Однако ранние процессоры выполняют операции с квадратным корнем из приведенных выше алгоритмов программными средствами, которые имеют большие задержки для его завершения.

# 1 Обзор методов вычисления квадратного корня

С быстрым развитием технологии, которая позволяет интегрировать большие схемы в одном кристалле, а также увеличить потребность в более быстром времени выполнения вычислений, аппаратная реализация операции с квадратным корнем стала более привлекательной. К сожалению, из-за сложности алгоритмов квадратного корня вычисление квадратного корня нелегко осуществить с помощью технологии ПЛИС (FPGA). Есть несколько алгоритмов квадратного корня, которые реализованы на FPGA. Как правило, они сгруппированы в две разные категории. В первой категории называются методы оценки, такие как грубая оценка и метод Ньютона-Рафсона (а также его производное: CORDIC), а во вторую категорию - метод цифра за цифрой.

Наконец, необходимо классифицировать дальнейший метод «цифра за цифрой» на два отдельных класса: алгоритм с восстановлением и без восстановления. Первоначально, алгоритм с восстановлением проложил путь для всех других методов, но сейчас он утратил свою важность и в настоящее время больше не используется. По сравнению с ним, алгоритм без восстановления не восстанавливает остаток, и он может быть реализован с наименьшим количеством аппаратных ресурсов, и в результате получается простая аппаратная реализация. Он наиболее подходит для реализации FPGA и позволяет легко осуществлять стандартное округление.

Существует много стратегий или архитектур для реализации алгоритма без восстановления для преобразования квадратного корня в цифровую форму в аппаратном обеспечении ПЛИС. Алгоритм без восстановления можно реализовать как с полностью конвейерной, так и с итеративной версией, которая не требует ни умножителей, ни мультиплексоров. Многие усилия, которые предпринимаются, служат для уменьшения потребляемого оборудования с умеренной задержкой. Тем не менее, FPGA очень подходит для принятия полностью конвейерной архитектуры из-за особенностей своей структуры. Следовательно, очень мало лишних затрат, если конвейерная технология реализована в FPGA.

# 2 Метод Ньютона-Рафсона

Нахождение решения для системы нелинейных уравнений f (x) = (f1, ....... fn)' = 0 было проблемой в последние годы. Можем рассмотреть это нелинейное уравнение и попытаться найти его решение, и это можно решить методом Ньютона-Рафсона. Этот метод очень известен своей быстрой сходимостью и улучшением свойства сходимости.

Метод Ньютона очень быстр и эффективен по сравнению с другими методами. Поэтому для сравнения производительности очень важно соблюдать стоимость и скорость сходимости. Метод Ньютона требует только одну итерацию и оценку производной на одну итерацию. Результат сравнения скорости сходимости методов Бисекции, Ньютона и Секущих получился как метод Бисекции <метод Ньютона <метод Секущих, который по количеству состоит в том, что метод Ньютона в 7,678622465 раз лучше, чем метод Бисекции, тогда как метод Секущих в 1,338482397 раз лучше, чем метод Ньютона.

Сложные системы с более высокой скоростью управления обработкой пользуются спросом, и решение этой проблемы состоит в том, чтобы разделить их на подсистемы, и таким образом каждая подсистема будет обрабатываться индивидуально, а управление и эксплуатация будут применяться к каждой из этих подсистем.

Нахождение корней нелинейного уравнения с помощью метода Ньютона Рафсона дает хороший результат с быстрой скоростью сходимости, и МатЛаб также принял этот метод для нахождения корней, и инструмент, используемый для таких расчетов, является научным калькулятором.

Метод Ньютона (также известный как метод Ньютона-Рафсона), названный в честь Исаака Ньютона и Джозефа Рафсона, представляет собой метод оценки последовательных наилучших приближений к извлечению (или нулям) вещественной функции. Любой метод нахождения нуля (метод деления пополам, метод ложного положения, Ньютон-Рафсона и т.д.) Также можно использовать для нахождения минимума или максимума такой функции путем нахождения нуля в первой производной функции, если взять метод Ньютона как алгоритм оптимизации.

Идея метода Ньютона-Рафсона заключается в следующем: начинается с предварительной гипотезы, которая логически безопасна для истинного корня, затем цель аппроксимируется его линией отступления (которая может быть вычислена с использованием инструментов исчисления), и одна вычисляет x-отрезок этой линии отступления (что легко делается с помощью простой алгебры). Этот x-отрезок, как правило, будет более приближенным к корню функции, чем исходное предположение, и метод может быть повторен. На основе коллинеарного масштабирования и локального квадратичного приближения квазиньютоновские методы были улучшены, поскольку значение функции не используется полностью. Используя локальное квадратичное приближение, представлен улучшенный алгоритм для усиления устойчивости и доказана глобальная сходимость алгоритма.

(1)

Метод Ньютона-Рафсона по одной переменной реализуется следующим образом: учитывая функцию ƒ, определенную по вещественным числам x и ее производной ƒ ', мы начнем с первого предположения x0 для корня функции f. При условии, что функция удовлетворяет всем предположениям, сделанным при выводе формулы, лучшее приближение x1 является

(2)

Геометрически, (x1, 0) - пересечение с осью касательной к графику f в точке (x0, f (x0)). Процесс повторяется до тех пор, пока не будет достигнуто достаточно точное значение.

(3)

Метод Ньютона является одним из многих методов для вычисления квадратных корней. Рассмотрим пример нахождения квадратного корня числа.

Найдем квадратный корень из 612, это эквивалентно поиску решения для

(4)

Функция, которая используется в методе Ньютона и ее производная:

(5)

(6)

Инициализируем х0 из формулы Ньютона-Рафсона, например, числом 10 и в итоге получаем следующую последовательность итераций данного метода и конечный результат.

Где правильные цифры выделены жирным шрифтом. Всего за несколько итераций можно получить решение с точностью до десятичных знаков.

Немного поясню суть выбора х0: если х0 выбирается наиболее подходящее к искомому корню числа, то количество итераций будет минимальным, иначе, если х0 выбирается любое, допустим единицей, то количество итераций значительно увеличиться, но в конечном счете корень числа будет найден. Максимально выгодно брать А/2, где А – это число, корень которого будет вычисляться.

Метод Ньютона-Рафсона является всеобъемлющим для решения неквадратных и нелинейных задач. Скорость сходимости данного метода является быстрой по сравнению с некоторыми методами.

# 3 Обзор метода цифра за цифрой

В методе вычисления цифра за цифрой каждая цифра квадратного корня находится в последовательности, где на каждой итерации генерируется только одна цифра квадратного корня. Он имеет несколько преимуществ, таких как: каждая цифра найденного корня верна и ее не нужно будет менять позже; нахождение квадратного корня будет прекращено после того, как будет найдена последняя цифра; и алгоритм работает для любой числовой базы.

В целом, этот метод может быть разделен на два класса, алгоритм с восстановлением и без восстановления цифр.

В алгоритме с восстановлением процедура состоит из извлечения полученного квадратного корня, добавления к нему значения 01 и вычитания его с должным смещением из текущего остатка. 0 в 01 соответствует умножению на 2; 1 – новый предположительный бит. Новый разработанный корневой бит действительно равен 1, если результирующий остаток является положительным, и наоборот равен 0, и этот остаток должен быть восстановлен путем добавления только что вычтенного количества.

В алгоритме без восстановления не восстанавливается вычитание, если результат был отрицательным. Вместо этого он добавляет 11 к корню, разработанному до сих пор, и на следующей итерации выполняет сложение. Если сложение вызывает переполнение, то на следующей итерации вы возвращаетесь в режим вычитания.

На рис. 1.1 и 1.2 приведен пример для получения двоичного квадратного корня из 01011101 (эквивалентно десятичному значению 93) алгоритмом с восстановлением и без соответственно.

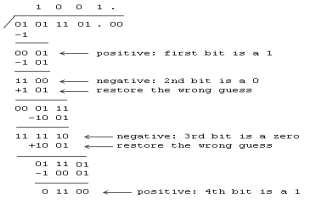


Рисунок 1.1 - Алгоритм с восстановлением

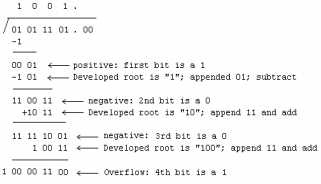


Рисунок 1.2 - Алгоритм без восстановления

Немного отличающийся от обычного не восстанавливающего алгоритма цифра за цифрой на рисунке 1.2 и модификация, показанная на рисунках 1.3, 1.4, может быть проведена для упрощения реализации и более быстрого вычисления. В этой модификации он использует только операцию вычитания и добавление 01, тогда как операция сложения и добавление 11 не используется. Он является эффективной версией алгоритма без восстановления квадратного корня.

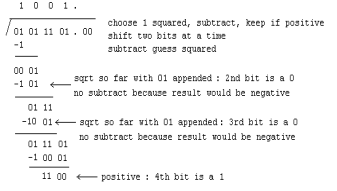


Рисунок 1.3 – Модифицированный алгоритм без восстановления

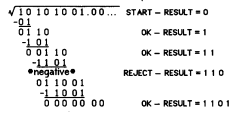


Рисунок 1.4 – Модифицированный алгоритм без восстановления

Алгоритм, предложенный для оценки квадратного корня, представляет собой двоичную версию хорошо известного, но часто упускаемого из употребления метода. В десятичной системе это довольно утомительный процесс, который чем-то напоминает восстановление деления, за исключением того, что две цифры берутся из подкоренного числа для каждой отдельной цифры, полученной в результате. К счастью, двоичная версия алгоритма очень проста, поскольку каждый новый бит в результате (0 или 1) выбирается путем простого сравнения.

Его можно сгруппировать с классом «прямых методов», который, как было установлено, лучше подходит для реализации в аппаратном обеспечении, чем традиционный метод Ньютона-Рафсона, которые был рассмотрен ранее.

# 4 Вавилонский метод (Babylonian method)

Вавилоняне разработали замечательный итерационный алгоритм для вычисления квадратных корней примерно в 1500 году до нашей эры. Этот алгоритм прост для понимания и удивительно быстр, он дает точность около 26 десятичных знаков всего за пять итераций. Алгоритм появился во вводных текстах по компьютерному программированию, а также в текстах для предварительного расчета, исчисления и численного анализа.

Чтобы приблизить , начнем с начального приближения, = x. Из этого значения можно установить второе приближение так, чтобы ≈ A. Из этого обязательно следует, что путем усреднения двух значений достигается второе и лучшее приближение. Пусть это приближение равно .

Этот метод можно понять, оценив его с помощью случаев:

Случай 1: . Таким образом, новое приближение становится И все готово.

Случай 2: . В этом приближении и, следовательно, Таким образом, новое приближение будет ближе к x или к .

Случай 3: . Для этого случая .

Таким образом, мы установили метод, изложенный вавилонянами, но на самом деле мы не установили его обоснованность. Каким образом сходится серия итераций, если она вообще существует? Для этого обратимся к древним грекам, которые признали, что среднее арифметическое двух чисел a, b всегда больше или равно среднему геометрическому: a + b ≤ . Таким образом, новое приближение всегда будет больше (или равно), чем (и только равно, когда a = b). Но что в каждом последующем приближении? Взяв и применив метод снова, мы получим . Как мы показали, ≥ , так ≤ . Поскольку ≤ , имеем следующее ≤ , откуда следует что . Продолжая это, мы видим, что для данного n (действительно, достаточно большое n: начальное значение может быть больше, чем ), . Аналогично, для достаточно больших n, . (Оба неравенства могут быть доказаны с помощью индукции.)

Сходимость метода не должна быть очень удивительной. В общем, вавилонский метод сходится относительно быстро. Для , используя оценку x = 1.5, мы имеем = 1.75, что уже примерно на 0,02 от фактического значения = 1.73.

Вавилонский метод действительно сходится, но как быстро он сходится? Чтобы исследовать этот вопрос, мы должны сначала установить значение скорости сходимости.

Начнем с определения функции ƒ для вавилонского метода, такой что . Вавилонский метод, конечно, итеративен, но наблюдение скорости изменения этой функции при очень хороших оценках может позволить нам увидеть, как функция сходится в определенных семенах.

Для этой корневой функции приближения мы можем наблюдать скорость изменения в любой точке приближения, беря производную от ƒ в этой точке. С нашей корневой функцией приближения имеем . Рассматривая графики (рисунок 1.5 и 1.6) для ряда функций, можно сделать несколько выводов:

:

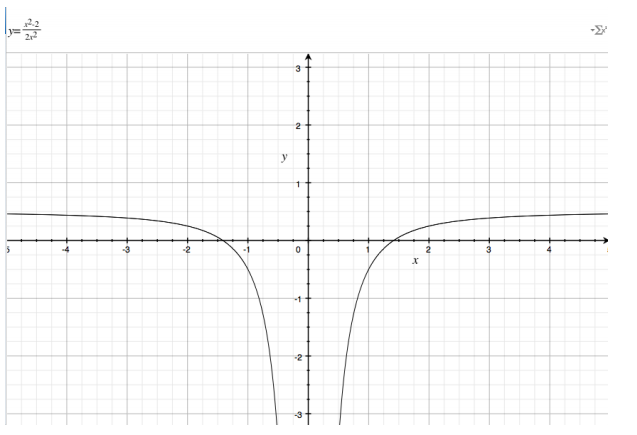


Рисунок 1.5 – Сходимость функции для

:

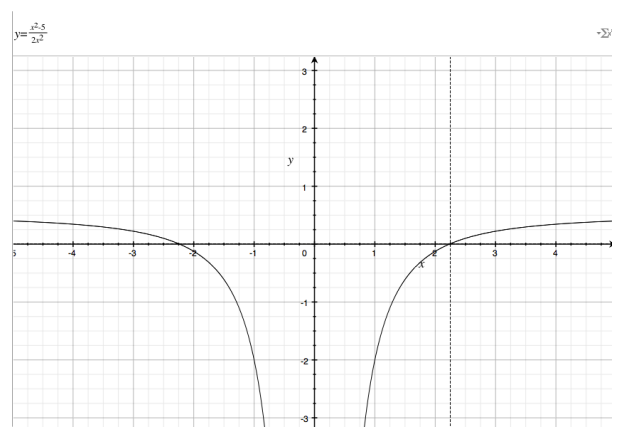


Рисунок 1.6 – Сходимость функции для

Для обоих графиков, как и в случае любого графика производной вавилонской функции в данном корне, местоположение корня находится точно при , что может проверить простая алгебра. Это конечно предоставляет удобный способ графической оценки квадратных корней, но также ставит некоторые вопросы. Одним из важных вопросов является связь между производной функцией и скоростью сходимости фактического итерационного метода.

Конечно, поскольку начальное число √A будет непрерывно генерировать себя в итеративном процессе, скорость изменения равна 0. Поведение производной функции вокруг неподвижной точки сильно отличается. Используя √8 в качестве модели, видно, что количество итераций, необходимое для схождения различных начальных чисел, не напрямую связано с производной, а моделируется ее поведением. Для начального числа x = 2 требуется примерно 4 итерации, чтобы получить приближение с точностью до 7 десятичных знаков. Поскольку ƒ '(2) = - 1/2, можно ожидать, что другое семя с различной скоростью изменения будет сходиться либо быстрее, либо медленнее.

Немного рассмотрим вавилонский метод нахождения квадратного корня числа А. Мы начнем с предположения об А и назовем его . На рисунке 1.7 показан случай, когда .

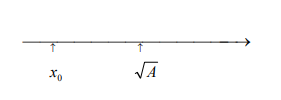


Рисунок 1.7 – Отмеченные и на прямой

Поскольку меньше , то и, следовательно, будет больше , как показано ниже на рисунке 1.8.

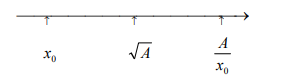


Рисунок 1.8 – Отмеченные , , на прямой

Поскольку находится между двумя значениями, представляется разумным попытаться в нашем следующем приближении усреднить эти два значения. Таким образом мы пытаемся

(7)

использовать наше второе приближение к , как показано на рисунке 1.9.

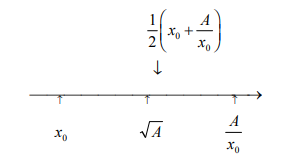


Рисунок 1.9 – Приближенное значение к

Продолжая таким образом, мы берем наше третье приближение и теперь имеем общее рекурсивное соотношение

для всех n = 0, 1, 2, 3, … (8)

Прошлое соотношение называется вавилонским алгоритмом для .

Это первый метод приблизительного расчета квадратного корня неотрицательного действительного числа. В этом методе сначала определяются два значения: одно - завышение квадратного корня действительного неотрицательного числа, а другое - по оценке. Среднее из этих двух должно обеспечить результат. Но для этого нужны арифметические и геометрические средства, которые показывают, что среднее значение всегда является завышенным значением квадратного корня. Некоторая ошибка вычисления также возникает во время вычисления, и это не может быть вычислено точно. Таким образом, это не эффективный метод вычисления квадратного корня вещественного неотрицательного числа.

# 5 Разбор метода цифра за цифрой

Предыдущие алгоритмы квадратного корня обеспечивают сложный способ получения точного результата и неэффективны для реализации в FPGA. Не восстанавливающий алгоритм - эффективный способ вычисления квадратного корня из действительного числа. Это не использует больше числа арифметических операций. Таким образом, для определения результата требуется меньше времени на вычисления и выполнение. Это также обеспечивает меньшую сложность, а также точный результат. Для не восстановительного алгоритма вычисления квадратного корня не требуется множитель. Этот алгоритм наиболее подходит для реализации FPGA схемы квадратного корня. Предложенный алгоритм пропускает этапы восстановления и дополнительного сложения алгоритма восстановления.

# 6 Расчет частотно-пространственной характеристики чувствительности

# 4 ТЕХНИКО-ЭКОНОМИЧЕСКОЕ ОБОСНОВАНИЕ ВКР

Выпускная квалификационная работа посвящена обзору методов извлечения квадратного корня на ПЛИС, сравнении и реализации наилучшего метода. Для реализации выбранного метода воспользовались ПЛИС (Программируемая Логическая Интегральная Схема) и математическим описанием методов.

В этой ВКР были изучены технология и методы извлечение квадратного корня, как в математическом смысле, так и в аппаратном. Эта тема является актуальной в связи с ростом количества информации и, соответственно, скоростью ее обработки, сейчас эта проблема является одной из самых важных.

**4.1. Подсчет расходов на выполненную работу**

Себестоимость данной выпускной работы будет рассчитываться по фактическим затратам. В расчет себестоимости работы включаются следующие пункты:

1. Подсчет расходов на оплату труда;

2. Расчет накладных расходов;

3. Расходы на материалы;

4. Отчисления на социальные нужды;

5. Прочие прямые расходы;

**4.1.1** **Подсчет расходов на оплату труда**

На основе трудоемкости выполнения работы по анализу и имплементации методов извлечения квадратного корня, вычисляются издержки на оплату труда ее исполнителей, являющиеся главным пунктом в себестоимости разработки.

Для вычисления затрат на этапе проектирования необходимо определить трудоемкость каждой работы. Продолжительность работ определяется в человеко-днях. Ставка заработной платы в день определяется как отношение средней заработной в месяц по данным к числу рабочих дней в месяце (21 день):

По данным веб-ресурса russia.trud.com состояние заработных плат у инженеров-программистов. В итоге получилось, на 20.05.19, по профессии инженер-программист в России открыто 982 вакансий. Для 42.6% открытых вакансий, работодатели указали заработную плату в размере 2 600 – 49 500 руб., 37% объявлений, с зарплатой 49 500 – 96 400 руб., и 12.8% с зарплатой 96 400 - 143 000 руб.

Допустим, у инженера-программиста заработная плата будет составлять 28700 руб., так как опыта работы, либо нет, либо минимальный. Тогда заработная плата в день будет равна:

Просмотрев приложение 1 приказа «Размеры должностных окладов ППС с учетом повышающих квалификационных коэффициентов» от 07.03.2019 №ОД/0097, заработная плата доцента с ученой степенью кандидата наук будет соответствовать 37 700 рублей.

Следовательно, заработная плата в день будет равна:

План работ студента (инженера-программиста), выполняющего ВКР, и научного руководителя представлен в таблице 4.1. и таблице 4.2.

Таблица 4.1 – План работ разработчика

|  |  |  |
| --- | --- | --- |
| Этапы и содержание выполняемых работ | Трудоемкость, человек/день. | Ставка, руб./день. |
| Обзор литературы и составление ТЗ | 4 | 1367 |
| Анализ ТЗ и предметной области | 5 | 1367 |
| Работа с различными источниками | 3 | 1367 |
| Изучение методов извлечения квадратного корня | 3 | 1367 |
| Имплементация метода извлечения квадратного корня на ПЛИС | 12 | 1367 |
| Консультации с научным руководителем | 4 | 1367 |
| Тестирование метода | 2 | 1367 |
| Создание технико-экономического обоснования | 2 | 1367 |
| Оформление пояснительной записки | 10 | 1367 |
| Сдача работы | 1 | 1367 |

Таблица 4.2 – План работ руководителя

|  |  |  |
| --- | --- | --- |
| Этапы и содержание выполняемых работ | Трудоемкость, чел/день. | Ставка, руб./день. |
| Разработка ТЗ | 1 | 1796 |
| Анализ ТЗ и предметной области | - | 1796 |
| Тестирование метода | 2 | 1796 |
| Консультации с научным руководителем | 4 | 1796 |
| Сдача проекта | 1 | 1796 |

Определение заработной платы исполнителей, которые напрямую связаны с данной работой) происходит по трудоемкости работ (таблица 4.1 и таблица 4.2), связанных с выполнением ВКР.

Основная и дополнительная заработная плата исполнителей, т.е. научного руководителя и инженера-программиста (студента), выполняющего дипломную работу, рассчитываются на основании ниже приведенных данных:

• Трудоемкость работ инженера-программиста по таблице 4.1

• Трудоемкость работ научного руководителя по таблице 4.2

• Ставка инженера-программиста и руководителя указаны в таблице 4.1 и таблице 4.2 соответственно

• Норму дополнительной заработной платы примем в 8,3%

• Норму отчислений на социальные нужды примем в 29,8%

Основная заработная плата исполнителей данной работы вычисляется по формуле:

Дополнительная заработная плата соответствует 8,3% от основной:

Отчисления на социальные нужды соответствуют 29,8% от суммы основной и дополнительной заработной платы:

**4.1.2 Расчет накладных расходов**

Накладные расходы в приведенной работе определяются как 36% от суммы основной и дополнительной заработных плат:

**4.1.3 Расходы на материалы**

Расчет количества и стоимости материалов с учетом транспортно-заготовительных расходов: в пункт «Материалы» входят расходы на основные и дополнительные материалы, которые, конечно, будут необходимы для выполнения данной выпускной работы.

Вычисление расходов по пункту «Материалы» представлены далее в таблице 4.3. Стоимость материалов, использованных в данной работе, была взята с веб-ресурса магазина канцелярских товаров «Комус» <https://www.komus.ru/katalog/bumaga-i-bumazhnye-izdeliya/bumaga-dlya-ofisnoj-tekhniki/formatnaya-bumaga/c/135000/?from=menu>

Таблица 4.3 – расходы пункта «Материалы»

|  |  |  |  |
| --- | --- | --- | --- |
| Материалы | Кол-во | Цена, руб. | Сумма, руб. |
| Блок листов А4 для печати | 1 | 296 | 296 |
| Печать необходимого материала (картридж Colortek 725) | 1 | 500 | 500 |
| Канцелярские товары |  |  | 200 |
| ИТОГО: | | | 1196 |
| Транспортные расходы (15%) | | | 179,4 |
| ВСЕГО: | | | 1175,4 |

Данные по картриджу брались из веб-ресурса <https://www.ulmart.ru/goods/4208708>

**4.1.4 Издержки на амортизацию ПК**

Амортизация оборудования (в нашем варианте это ПК, ПЛИС и принтер) вычисляется по государственным нормам. За целый год сумма амортизации будет соответствовать:

где стоимость оборудования; норма амортизации

Стоимость ноутбука на данный момент такого ноутбука в продаже уже нет, но 5 лет назад его стоимость была 33600 руб.

Вычислим норму амортизации. Опираясь на постановление правительства РФ от 01.01.2002 №1 (с внесенными в него поправками от 28.04.2018 №256) "О Классификации основных средств, включаемых в амортизационные группы", машины офисные, в подгруппу которых входят персональные компьютеры, относятся ко второй группе, и, следовательно, к имуществу со сроком действия от двух до трех лет. Будем считать, что оборудование испытанное и исправное и возьмем средний срок два года.

Тогда получим:

Вычислим годовую сумму амортизации с учетом того, что использовался ноутбук:

За рабочий день сумма амортизации будет составлять:

где количество рабочих дней (247 дней на 2019 год)

Амортизация оборудования, использованного в данной работе, за все время разработки выпускной работы равна:

**4.1.5** **Прочие прямые расходы**

Прочие прямые расходы, включающие в себя затраты на использование средств связи (интернет), приведены в таблице 4.4.

Таблица 4.4 – прочие прямые расходы

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование | Кол-во | Цена, руб. | Сумма, руб. |
| Абонентская плата за интернет, месяц | 2 | 480 | 960 |
| ИТОГО: | | | 960 |

**4.1.6. Себестоимость работы над дипломным проектом**

Таблица 4.5 – Общие затраты

|  |  |
| --- | --- |
| **Пункт затрат** | **Сумма, руб.** |
| Материалы | 1175,4 |
| Расходы на оплату труда | 79040 |
| Дополнительная заработная плата |  |
| Отчисления на социальные нужды |  |
| Издержки на амортизацию ПК | ,4 |
| Прочие прямые расходы | 960 |
| Накладные расходы |  |
| ИТОГО: | 147268,8 |

Себестоимость выполнения ВКР целиком и полностью представлена в таблице 4.5

В итоге мы получаем, что полная себестоимость разработки данной работы составит: 147268 руб. 80 коп.

**4.2. Выводы**

В данном разделе был произведен подсчет и анализ с экономической составляющей выпускной работы. Подсчитана себестоимость данного продукта со стороны вложения в него финансовых средств.

Производя анализ основных затрат, мы видим, что максимальные расходы на выполнение данного проекта приходятся на основную заработную плату исполнителям этой работы.

В расчет себестоимости этой работы не включались средства, предоставляемые ВУЗом, таких как ПЛИС и т.д.

В итоговом счете, опираясь на представленные расчёты, себестоимость продукта, разрабатываемым в данной выпускной работе составляет: 147268 руб. 80 коп.

# Заключение

# Список литературы