###### МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

###### ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

###### НОВОСИБИРСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

###### Факультет информационных технологий

**Кафедра параллельных вычислений**

ОТЧЕТ

О ВЫПОЛНЕНИИ ЛАБОРАТОРНОЙ РАБОТЫ

«ВВЕДЕНИЕ В АРХИТЕКТУРУ ARM»

студента Бородина Артёма Максимовича 2 курса, 19205 группы

Направление 09.03.01 – «Информатика и вычислительная техника»

Преподаватель:

к.т.н, доцент

А.Ю. Власенко

Новосибирск 2020

**СОДЕРЖАНИЕ**

[ЦЕЛЬ 3](#_Toc18443921)

[ЗАДАНИЕ 3](#_ЗАДАНИЕ)

[ОПИСАНИЕ РАБОТЫ 4](#_Toc18443923)

[ЗАКЛЮЧЕНИЕ 5](#_Toc18443924)

[Приложение *1.**Листинг 1* 6](#_Приложение_1._)

[Приложение *2.*](#_Приложение_2._)*Ассемблерный код (ключ -O0)* 7

[Приложение *3.*](#_Приложение_3._)*Ассемблерный код (ключ -O2)* 8

# ЦЕЛЬ

1. Знакомство с программной архитектурой ARM.

2. Анализ ассемблерного листинга программы для архитектуры ARM.

# ЗАДАНИЕ

*1 вариант:*

Алгоритм вычисления числа π с помощью разложения в ряд (ряд Грегори-Лейбница) по формуле Лейбница N первых членов ряда.

1. Изучить основы программной архитектуры ARM.

2. Для программы на языке Си (из лабораторной работы 1) сгенерировать ассемблерные листинги для архитектуры ARM, используя различные уровни комплексной оптимизации.

3. Проанализировать полученные листинги

4. Составить отчет по лабораторной работе.

# ОПИСАНИЕ РАБОТЫ

1. Изучены основы архитектуры ARM, группы регистров, доступные программисту, сопроцессор и векторные расширения.
2. Были сделаны 3 листинга: код программы, вычисляющей число π, код программы на ассемблере, скомпилированной с ключом -O0 и код программы на ассемблере, скомпилированной с ключом -O2.
3. Для просмотра кода программы на ассемблере был использован ресурс godbolt.org. Компиляция проводилась для ARM64. (Листинги можно посмотреть по ссылке: https://godbolt.org/z/9nz6Y6)
4. При сравнении двух листингов на ассемблере становится заметно, что программа, скомпилированная с ключом -O2, требует гораздо меньше памяти для переменных и делает расчеты более эффективно (например, использует регистры для более быстрого доступа к данным или избегает хранения ненужных значений в случае с clock), по сравнению с программой, скомпилированной с ключом -O0. Так же компилятор с ключом -O2 добавил проверку на 0 в функции **LeibnizFormula(int)**, что позволило избежать заход в цикл для расчета числа π, т.к. после инициализации переменных цикл пришлось бы сразу завершать.

# ЗАКЛЮЧЕНИЕ

После изучения архитектуры ARM, были рассмотрены программы на ассемблере, скомпилированные с ключами -O0 и -O2, и проведено последующее сравнение для поиска различий разных уровней оптимизации.

# Приложение 1. *Код программы*



# Приложение 2. *Ассемблерный код (ключ -O0)*



# Приложение 3. *Ассемблерный код (ключ -O2)*

