

**Министерство образования и науки Российской Федерации Московский
физико-технический институт (государственный университет)**

**Физтех-школа радиотехники и компьютерных технологий
Кафедра Микропроцессорных технологий в интеллектуальных
системах управления
Syntacore**

Выпускная квалификационная работа бакалавра

Гибкий подход к подъёму LLVM MIR кода открытой архитектуры RISC-V в SSA форму LLVM IR

Автор:

Студент Б01-110 группы
Романов Александр Викторович

Научный руководитель:

Владимиров Константин Игоревич



Москва 2025

Аннотация

Гибкий подход к подъёму LLVM MIR кода открытой архитектуры RISC-V в SSA форму LLVM IR

Романов Александр Викторович

Проблема бинарной совместимости программ и их переносимости на разные архитектуры без возможности перекомпиляции часто решается при помощи бинарных трансляторов. Существует большое количество статических и динамических бинарных трансляторов. Большинство из них работают либо за счёт прямого сопоставления инструкциям и регистрам исходной архитектуры инструкции и регистры целевой архитектуры, либо за счёт паттерн матчинга. Такие решения делают сложным поддержание новых исходных архитектур ввиду чего поддержка относительно молодой микропроцессорной архитектуры RISC-V в существующих трансляторах либо отсутствует, либо сильно ограничена.

В данной работе рассмотрен новый инструмент для подъёма машинно зависимого представления RISC-V кода LLVM MIR в высокоуровневое машинно-независимое представление LLVM IR и его применение для простой статической трансляции бинарного RISC-V кода на любую поддерживаемую LLVM архитектуру.

Contents

1	Introduction	1
1.1	Problem	1
1.2	Motivation	1
1.3	Objectives	1
1.4	Outline	1
2	Background	1
2.1	e.g. User Feedback	1
2.2	e.g. Representational State Transfer	1
2.3	e.g. Scrum	1
3	Related Work	1
4	Requirements	1
4.1	Overview	1
4.2	Existing System	1
4.3	Proposed System	1
4.3.1	Functional Requirements	1
4.3.2	Quality Attributes	1
4.3.3	Constraints	1
4.4	System Models	1
4.4.1	Scenarios	1
4.4.2	Use Case Model	1
4.4.3	Analysis Object Model	1
4.4.4	Dynamic Model	1
4.4.5	User Interface	2
5	Architecture	2
5.1	Overview	2
5.2	Design Goals	2
5.3	Subsystem Decomposition	2
5.4	Hardware Software Mapping	2
5.5	Persistent Data Management	2
5.6	Access Control	2
5.7	Global Software Control	2
5.8	Boundry Conditions	2

6 Case Study / Evaluation	2
6.1 Design	2
6.2 Objectives	2
6.3 Results	2
6.4 Findings	2
6.5 Discussion	2
6.6 Limitations	2
7 Summary	2
7.1 Status	2
7.1.1 Realized Goals	2
7.1.2 Open Goals	2
7.2 Conclusion	3
7.3 Future Work	3
List of Figures	4
Appendix A: Supplementary Material	5
Bibliography	6

1 Introduction

1.1 Problem

1.2 Motivation

1.3 Objectives

1.4 Outline

2 Background

2.1 e.g. User Feedback

2.2 e.g. Representational State Transfer

2.3 e.g. Scrum

3 Related Work

4 Requirements

4.1 Overview

4.2 Existing System

4.3 Proposed System

4.3.1 Functional Requirements

4.3.2 Quality Attributes

4.3.3 Constraints

4.4 System Models

4.4.1 Scenarios

4.4.2 Use Case Model

4.4.3 Analysis Object Model

4.4.4 Dynamic Model

4.4.5 User Interface

5 Architecture

5.1 Overview

5.2 Design Goals

5.3 Subsystem Decomposition

5.4 Hardware Software Mapping

5.5 Persistent Data Management

5.6 Access Control

5.7 Global Software Control

5.8 Boundry Conditions

6 Case Study / Evaluation

6.1 Design

6.2 Objectives

6.3 Results

6.4 Findings

6.5 Discussion

6.6 Limitations

7 Summary

7.1 Status

7.1.1 Realized Goals

7.1.2 Open Goals

7.2 Conclusion

7.3 Future Work

List of Figures

Appendix A: Supplementary Material

– Supplementary Material –

Bibliography

- [1] Marcus Aurelius, “Meditations.”