

ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

Факультет компьютерных наук

Образовательная программа «Программная инженерия»

СОГЛАСОВАНО

Доцент базовой кафедры

«Системное программирование»

Института системного программирования

им В.П. Иванникова РАН (ИСП РАН)

факультета компьютерных наук,

кандидат физико-математических наук

УТВЕРЖДАЮ

Академический руководитель

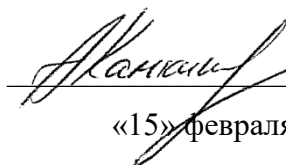
образовательной программы факультета


компьютерных наук

«Программная инженерия»

профессор департамента программной

инженерии, кандидат технических наук

 А. С. Камкин
«15» февраля 2023 г.

 В. В. Шилов
«7» апреля 2023 г.

ПРОТОТИП ОНЛАЙН-ГЕНЕРАТОРА ТЕСТОВЫХ ПРОГРАММ ДЛЯ
МИКРОПРОЦЕССОРОВ

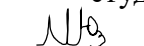
Техническое задание

ЛИСТ УТВЕРЖДЕНИЯ

RU.17701729.04.04-01 ТЗ 01-1-ЛУ

Исполнитель

студент группы БПИ201

 / М. Ю. Литвинов /
«31» января 2023 г.

Москва, 2023

Подп. и дата	
Инв. № дубл.	
Взам. инв. №	
Подп. и дата	
Инв. № подл	

**ПРОТОТИП ОНЛАЙН-ГЕНЕРАТОРА ТЕСТОВЫХ ПРОГРАММ ДЛЯ
МИКРОПРОЦЕССОРОВ**

Техническое задание

RU.17701729.04.04-01 ТЗ 01-1-ЛУ

Листов 20

СОДЕРЖАНИЕ

1.	ВВЕДЕНИЕ.....	4
	Наименование программы	4
	Наименование программы на английском языке.....	4
	Краткая характеристика области применения	4
2.	ОСНОВАНИЯ ДЛЯ РАЗРАБОТКИ	6
	Документы, на основании которых ведется разработка	6
3.	НАЗНАЧЕНИЕ РАЗРАБОТКИ.....	7
	Функциональное назначение	7
	Эксплуатационное назначение	7
4.	ТРЕБОВАНИЯ К ПРОГРАММЕ	8
	Требования к функциональным характеристикам	8
	Состав выполняемых функций	8
	Организация входных данных	8
	Состав описания ISA тестируемого прототипа.....	9
	Состав описания частично заданных и строго зафиксированных тестов	10
	Состав описания мутаций созданных последовательностей инструкций.....	11
	Организация выходных данных.....	11
	Требования к интерфейсу	11
	Требования к надежности	12
	Условия эксплуатации.....	12
	Требования к составу и параметрам технических средств	12
	Требования к информационной и программной совместимости	13
	Требования к маркировке и упаковке.....	13
	Требования к транспортировке и хранению.....	13
5.	ТРЕБОВАНИЯ К ПРОГРАММНОЙ ДОКУМЕНТАЦИИ.....	14
	Состав программной документации	14
	Специальные требования к программной документации	14
6.	ТЕХНИКО-ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ.....	15

Ориентировочная экономическая эффективность 15

Предполагаемая потребность..... 15

Экономические преимущества разработки по сравнению с отечественными и зарубежными аналогами 15

7. СТАДИИ И ЭТАПЫ РАЗРАБОТКИ 17

Подготовительный этап разработки (Ноябрь 2022)..... 17

Основной этап разработки (Декабрь 2022 – Март 2023) 17

Заключительный этап разработки (Апрель 2023 – Май 2023) 17

8. ПОРЯДОК КОНТРОЛЯ И ПРИЕМКИ..... 18

Виды испытаний..... 18

Общие требования к приемке работы..... 18

9. СПИСОК ИСПОЛЬЗОВАВШИХСЯ ИСТОЧНИКОВ 19

10. ПРИЛОЖЕНИЯ 20

Приложение 1 20

1. ВВЕДЕНИЕ

Наименование программы

Прототип онлайн-генератора тестовых программ для микропроцессоров.

Наименование программы на английском языке

Prototype of Online Test Program Generator for Microprocessors.

Краткая характеристика области применения

Программа, предоставляющая возможность автоматической генерации тестовых последовательностей из инструкций, составляющих целевую микропроцессорную архитектуру (Instruction Set Architecture, ISA) некоторого семейства микропроцессоров, а также возможности непосредственного исполнения сгенерированных тестовых последовательностей (динамическое/функциональное онлайн-тестирование) и сравнения результатов выполнения каждой тестовой последовательности с ее несколько измененной в рамках задаваемых правил версией.

Поставленная задача выполняется в рамках верификации как одного из этапов проектирования микропроцессоров и решение поставленной задачи имеет применение в автоматизации процесса системной верификации микропроцессоров[6], модели которых могут быть реализованы в сверхбольших интегральных схемах (СБИС), программируемых логических интегральных схемах (ПЛИС), а также запущены через программные эмуляторы. В решении задачи применяются методы случайной генерации входных данных, использующихся в инструкциях микропроцессоров в рамках генерируемых тестовых последовательностей.

В силу сложности масштабирования поставленной задачи по отношению к произвольно-задаваемым ISA, итоговое программное решение должно обеспечивать корректную работу как минимум в рамках микропроцессорной архитектуры RISC-V с базовым набором инструкций RV32I¹, однако подразумевается проектирование решения с учетом упомянутого масштабирования задачи в перспективе.

¹ Подразумевается базовый набор инструкций RV32I, спецификация которого актуальна на момент разработки[1].

Изм.	Лист	№ док.	Подп.	Дата
RU.17701729.04.04-01 ТЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

Поскольку процесс верификации микропроцессоров не обязан включать в себя избыточную настройку вспомогательных программных средств, взаимодействующих с верифицируемой реализацией (прототипом) модели микропроцессора (в частности, настройку операционной системы и средств компиляции пользовательских программ), и раз ISA могут варьироваться от реализации к реализации, то под каждую тестируемую реализацию необходимо предварительно осуществить сборку специфической версии разрабатываемого генератора тестовых последовательностей с помощью средств кросс-компиляции перед ее непосредственной выгрузкой на верифицируемый прототип, где итоговый генератор не должен опираться на код, взаимодействующий с операционной системой.

Изм.	Лист	№ док.	Подп.	Дата
RU.17701729.04.04-01 ТЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

2. ОСНОВАНИЯ ДЛЯ РАЗРАБОТКИ**Документы, на основании которых ведется разработка**

Учебный план подготовки бакалавров по направлению 09.03.04 «Программная Инженерия»
и утвержденная академическим руководителем программы тема курсового проекта.

Изм.	Лист	№ док.	Подп.	Дата
RU.17701729.04.04-01 ТЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

3. НАЗНАЧЕНИЕ РАЗРАБОТКИ

Функциональное назначение

Программа предоставляет возможности генерации и исполнения тестовых последовательностей из инструкций целевой микропроцессорной архитектуры (Instruction Set Architecture, ISA) верифицируемой системы (по умолчанию предоставляется реализация RISC-V RV32I-совместимой модели микропроцессора), а также вынесения вердикта о корректности работы использующихся в тестах инструкций на основе сравнения результатов выполнения двух функционально эквивалентных в рамках заданных правил последовательностей.

Эксплуатационное назначение

Программа может являться компонентом более обширного комплекса программных средств для верификации микропроцессоров, ускоряя работу разработчиков микропроцессоров посредством автоматической генерации тестовых последовательностей из инструкций целевой ISA (с их непосредственным исполнением) и соответствующих им входных данных по установленным заранее принципам.

В частности, программа может применяться в отношении программируемых логических интегральных схем (ПЛИС), настроенной на работу с целевой ISA, при этом самостоятельно реализуя необходимые ей сервисы операционной системы по работе с памятью, если таковые понадобятся.

Изм.	Лист	№ док.	Подп.	Дата
RU.17701729.04.04-01 ТЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

4. ТРЕБОВАНИЯ К ПРОГРАММЕ

Требования к функциональным характеристикам

Состав выполняемых функций

- Использование пользовательских описаний ISA тестируемой модели микропроцессора, моделей (шаблонов) частично заданных тестов и строго зафиксированных тестов, логики изменения (мутаций) генерируемых последовательностей инструкций, а также логики сравнения результатов выполнения оригинальной и измененной последовательностей инструкций для определения специфики конкретного генератора;
- Последовательная генерация последовательностей из инструкций целевой микропроцессорной архитектуры по предоставленным моделям частично заданных тестов с подготовкой требуемых для их полного определения входных данных;
- Последовательная генерация последовательностей из инструкций целевой ISA по описаниям строго зафиксированных тестов;
- Использование пользовательского описания логики мутаций последовательностей генерируемых инструкций для их изменения в границах исходного уровня функциональной эквивалентности и дальнейшего сравнения результатов выполнения каждой сгенерированной последовательности инструкций целевой ISA с ее измененной версией;
- Подготовка и вывод итоговых вердиктов об успешности выполнения тестируемым прототипом микропроцессора каждой сгенерированной последовательности инструкций и ее измененной версии по заданным правилам сравнения.

Организация входных данных

Так как работа с генератором в перспективе может производиться на bare-metal системах, каждая из которых обладает собственной ISA и моделью памяти, то уже на этапе компиляции требуется указать все те параметры и пользовательские определения, которые будут использоваться в скомпилированном онлайн-генераторе, при этом исключив необходимость обращений к файловой системе уже в процессе работы программы.

В первую очередь, пользователю необходимо передать информацию о кросс-компиляторе языка Си (стандарт C99) и передаваемых ему параметрах, а также используемую верифицируемой системой

Изм.	Лист	№ док.	Подп.	Дата
RU.17701729.04.04-01 ТЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

длину машинного слова, которые будут использоваться в процессе сборки исполняемого модуля генератора тестовых программ непосредственно под тестируемую модель микропроцессора.

Далее, пользователю требуется предоставить четыре абсолютных или относительных пути в файловой системе, описывающих расположение файлов описания ISA для тестируемой модели микропроцессора, файлов с описанием моделей частично заданных и строго зафиксированных тестов, файлов с описанием логики мутаций генерируемых последовательностей инструкций соответственно и ожидаемое расположение итогового файла онлайн-генератора.

Дополнительно пользователь может использовать флаг, сигнализирующий об использовании функции «printf», объявленной в заголовке «stdio.h» из стандартной библиотеки языка Си, для передачи итоговых вердиктов об исполнении тестовых последовательностей непосредственно в стандартный поток вывода.

Так как работа с генератором не обязательно предполагает наличие операционной системы на целевом микропроцессоре (подобное программное окружение принято называть bare-metal системами), а значит, не предполагается и доступ к стандартным средствам выделения динамической памяти, по возможности, все использующиеся массивы данных с информацией, необходимой для функционирования генератора, должны быть объявлены статически.

Отдельные части структуры предоставляемых пользователем файлов должны быть зафиксированными для корректного взаимодействия с ядром генератора - их описание должно быть включено в документ «Прототип онлайн-генератора тестовых программ для микропроцессоров». Руководство системного программиста (ГОСТ 19.503-79), в то время как общие требования к ним описываются ниже.

После непосредственной компиляции исполняемого модуля генератора тестовых программ он может быть загружен на исполнение в системе, подлежащей тестированию – на данном этапе никакие пользовательские входные данные уже не требуются.

Состав описания ISA тестируемого прототипа

Файлы с описанием ISA для верифицируемого прототипа должны включать в себя следующий минимальный набор аспектов:

- Описание количества регистров общего назначения, а также количества регистров для работы с вещественными числами, если те используются;

Изм.	Лист	№ док.	Подп.	Дата
RU.17701729.04.04-01 ТЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

- Описание количеств присутствующих в описании обычных инструкций и групп, на которые их можно поделить, а также псевдо-инструкций, реализуемых через имеющиеся обычные инструкции;
- Описание реализации зафиксированного интерфейса для способов выгрузки значений регистров из/в статически выделенные области оперативной памяти без изменения до/во время/после операции выгрузки соответствующих регистрам значений;
- Описание представлений групп инструкций, не-параметризованных инструкций, псевдо-инструкций и параметризованных инструкций;
- Описание реализации зафиксированного интерфейса для кодирования (ассемблирования) инструкции по их имеющимся данным;
- Описание реализации зафиксированного интерфейса для сравнения двух полученных в процессе выполнения наборов значений регистров – для исходной сгенерированной тестовой последовательности инструкций и ее измененной версии.

Состав описания частично заданных и строго зафиксированных тестов

Файлы с описанием частично заданных и строго зафиксированных тестов для верифицируемого прототипа должны включать в себя следующий минимальный набор аспектов:

- Описание максимальных допустимых количеств инструкций в рамках одной генерируемой последовательности (обязательный параметр) и инструкций, добавляемых в процессе мутаций последовательностей (необязательный параметр – при его отсутствии количество генерируемых дополнительно инструкций будет зависеть от первого упомянутого параметра);
- Описание количеств частично заданных и строго зафиксированных тестов;
- Описание частично заданных тестов через конкретный для всех таких тестов интерфейс;
- Описание строго зафиксированных тестов через конкретный для всех таких тестов интерфейс;
- Описание реализации зафиксированного интерфейса для заполнения инструкций по умолчанию в рамках одной тестовой последовательности, если для обрабатываемого шаблона не задавалась специфическая функция;

Изм.	Лист	№ док.	Подп.	Дата
RU.17701729.04.04-01 ТЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

Состав описания мутаций созданных последовательностей инструкций

Файлы с описанием мутаций (с сохранением функциональной эквивалентности) созданных последовательностей инструкций должны включать в себя реализацию интерфейса для добавления конкретного количества дополнительных строк в выделенную под это область памяти на основании сгенерированной ранее тестовой последовательности инструкций.

Организация выходных данных

На этапе компиляции выходными данными считаются статически-скомпилированный исполняемый файл самого онлайн-генератора тестовых программ, а также набор файлов использовавшихся при сборке.

На этапе исполнения в качестве основных выходных данных предоставляются вердикты об успешности/безуспешности корректности работы использовавшихся в тестах инструкций на основании сравнения результатов выполнения каждой сгенерированной тестовой последовательности инструкций и ее измененной версии, а также должны предоставляться некоторые дополнительные данные, необходимые для воспроизведения тестов с некорректным завершением работы – все вердикты и необходимые для их описания данные должны соответствовать некоторому общему интерфейсу и располагаться в оперативной памяти, связанной с верифицируемым прототипом.

Если при компиляции генератора использовался флаг использования стандартного потока вывода, описанный в пункте 4.1.2 «Организация входных данных», то информация о вердиктах выполнения тестовых последовательностей инструкций будет дополнительно выводиться в упомянутый стандартный поток вывода.

Требования к интерфейсу

В случае использования на этапе компиляции выделенного флага для использования стандартного потока вывода языка Си, упомянутого в пункте 4.1.2 «Организация входных данных», с целью текстового отображения итоговых вердиктов о выполнении конкретных тестовых последовательностей, сама визуализация будет осуществляться в окне командной строки (терминала) одной из указанных в пункте 4.5 «Требования к составу и параметрам технических средств» операционных систем – в противном же случае стандартное получение данных вердиктов о выполнении конкретных тестовых последовательностей из оперативной памяти тестируемой системы будет осуществляться через сторонние интерфейсы, не описываемые в рамках текущей программной реализации.

Изм.	Лист	№ док.	Подп.	Дата
RU.17701729.04.04-01 ТЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

Требования к надежности

За счет специфики поставленной в рамках программной реализации задачи и ее предполагаемого решения, преждевременное завершение работы программы возможно в случае несоответствий описанным в пункте 4.1.2 «Организация входных данных» правилам или в случае технических неполадок, не подлежащих исправлению со стороны работы программы (например, неисправность используемой в процессе тестирования ПЛИС).

В то же время, использование предоставленных вместе с исходным кодом проекта готовые описания ISA RISC-V RV32I и базовой логики совершения мутаций над последовательностями тестовых инструкций должны успешно работать в связке с ядром генератора.

В случае использования на этапе компиляции выделенного флага для использования стандартного потока вывода языка Си, упомянутого в пункте 4.1.2 «Организация входных данных», с целью текстового отображения итоговых вердиктов о выполнении конкретных тестовых последовательностей, по используемому стандарту языка Си C99 большинство ошибок в процессе работы программы будут отражены в упомянутом стандартном потоке вывода.

Условия эксплуатации

Условия эксплуатации описываются в составе документа «Прототип онлайн-генератора тестовых программ для микропроцессоров». Руководство системного программиста (ГОСТ 19.503-79).

Требования к составу и параметрам технических средств

Для успешной компиляции программы требуется следующий состав технических средств:

- Компьютер с одной из установленных операционных систем – Linux (x86-64) или Linux (ARM64);
- Кросс-компилятор языка Си (стандарт C99) с генерацией кода, совместимого с тестируемым прототипом микропроцессора на уровне его ISA;
- Система CMake[2] с открытым исходным кодом для кроссплатформенного описания процесса компиляции исходного кода на языке Си версии 3.16 или более поздней;
- Наличие не менее 100МБ свободного места на жестком диске компьютера.

Иные требования совпадают с системными требованиями указанных операционных систем.

Поскольку скомпилированный прототип генератора будет использовать статическую память в глобальной области видимости в зависимости от требований самого пользователя (описывается набором использующихся регистров в рамках тестируемой системы, количеством тестов и итераций для отдельных

Изм.	Лист	№ док.	Подп.	Дата
RU.17701729.04.04-01 ТЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

тестов, объемом памяти для выгрузки результатов отдельных тестов и так далее), итоговое минимальное ограничение на объем оперативной памяти не может быть сформулировано однозначно.

В случае запуска генератора на уровне bare-metal системы (например, с использованием ПЛИС), могут потребоваться условия совместимости аппаратного обеспечения с существующими программными решениями для выделения регионов оперативной памяти, используемой в генераторе, а также для возможности изучить выгруженные генератором данные вердиктов о выполненных тестах.

Если же запуск генератора будет производиться в рамках эмулятора микропроцессора с опциональным флагом вывода сопровождающих и отладочных сообщений, то вдобавок к вышеупомянутым требованиям добавится и требование совместимости с набором интерфейсов системных вызовов в рамках стандарта C99.

Требования к информационной и программной совместимости

Разработка и отладка кода ядра генератора тестовых инструкций для микропроцессоров, а также одной из реализаций описания ISA RISC-V RV32I, производится на языке программирования Си стандарта C99 с опциональным использованием заголовка стандартных операций ввода/вывода «stdio.h» стандартной библиотеки языка. Сама сборка производится через генератор систем сборки CMake с открытым исходным кодом.

Требования к маркировке и упаковке

Программа распространяется в виде электронного пакета, содержащего программную документацию, файлы исходного кода ядра генератора тестовых программ и частной реализации под ISA RISC-V RV32I, файл CMakeLists.txt с общим описанием процесса сборки исполняемого модуля генератора и иные необходимые для его функционирования и сборки файлы.

Требования к транспортировке и хранению

Программное изделие может храниться и транспортироваться как на физических носителях информации (например, USB-накопителях), так и в облачных хранилищах.

Изм.	Лист	№ док.	Подп.	Дата
RU.17701729.04.04-01 ТЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

5. ТРЕБОВАНИЯ К ПРОГРАММНОЙ ДОКУМЕНТАЦИИ

Состав программной документации

- «Прототип онлайн-генератора тестовых программ для микропроцессоров».
Техническое задание (ГОСТ 19.201-78);
- «Прототип онлайн-генератора тестовых программ для микропроцессоров».
Пояснительная записка (ГОСТ 19.404-79) ;
- «Прототип онлайн-генератора тестовых программ для микропроцессоров».
Руководство системного программиста (ГОСТ 19.503-79);
- «Прототип онлайн-генератора тестовых программ для микропроцессоров».
Программа и методика испытаний (ГОСТ 19.301-79);
- «Прототип онлайн-генератора тестовых программ для микропроцессоров».
Текст программы (ГОСТ 19.401-78).

Специальные требования к программной документации

Документы к программе должны быть составлены в соответствии с ГОСТ 19.106-78 и ГОСТами к каждому отдельному виду прилагаемых документов (раздел 5, «Состав программной документации»).

Пояснительная записка должна быть загружена в систему Антиплагиат через LMS «НИУ ВШЭ».

Документация и программа сдается в электронном виде в форматах .pdf или .docx в архиве форматов .zip или .rar .

За три дня до защиты проекта перед предметной комиссией все материалы курсового проекта (программная документация, программный проект, отзыв руководителя, отчет системы Антиплагиат) должны быть загружены одним или несколькими архивами в проект дисциплины «Курсовой проект» в личном кабинете в информационной образовательной среде SmartLMS НИУ ВШЭ.

Изм.	Лист	№ док.	Подп.	Дата
RU.17701729.04.04-01 ТЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

6. ТЕХНИКО-ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ

Ориентировочная экономическая эффективность

Использование автоматической генерации последовательностей тестовых инструкций для микропроцессоров позволит с некоторой точностью покрыть большинство возможных входных данных существующих инструкций, тем самым повышая итоговую надежность разрабатываемой модели микропроцессора.

Кроме того, использование подхода онлайн-генерации тестов в сравнении с оффлайн-генерацией позволит гораздо быстрее перейти к этапу непосредственного исполнения сгенерированных тестов.

В дополнение к этому утверждению, возможное использование разрабатываемого генератора в совокупности с ПЛИС позволит достичь таких скоростей исполнения отдельных инструкций, которые были бы сравнимы с аналогичными показателями на готовой СБИС на полупроводниковой подложке, что позволит получить наиболее точную характеристику разрабатываемой модели.

Предполагаемая потребность

На момент начала разработки не было выявлено примеров использования подхода онлайн-генерации последовательностей тестовых инструкций с целью верификации проектируемой RTL-модели микропроцессора. В дополнение к этому, проект разрабатывается по принципу открытого исходного кода, что предоставит возможности ручной адаптации существующего решения под специфические нужды в рамках верификации микропроцессоров.

Экономические преимущества разработки по сравнению с отечественными и зарубежными аналогами

С момента начала разработки известно о существовании несколько относительно схожих аналогов решения поставленной задачи (отечественных и зарубежных):

- ИСП РАН MicroTESK[4];
- IBM Threadmill[3];
- Common Hardware for Interfaces, Processors and Systems (CHIPS) RISC-V Generator [5].

Изм.	Лист	№ док.	Подп.	Дата
RU.17701729.04.04-01 ТЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

Все представленные аналоги реализуют подход оффлайн-тестирования, а потому не могут быть полностью сопоставлены разрабатываемому проектному решению, однако все же являются наиболее схожими по назначению.

Конкретно разработка Threadmill от IBM является закрытым программным решением, что противопоставляется текущему программному решению с открытым исходным кодом. Генератор тестовых программ от CHIPS реализован исключительно под RISC-V в отличие от подхода к обобщению под различные архитектуры в текущем программном решении. К тому же, данный аналог использует промежуточную симуляцию, что также сказывается на производительности в процессе выполнения последовательностей тестовых инструкций.

Изм.	Лист	№ док.	Подп.	Дата
RU.17701729.04.04-01 ТЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

7. СТАДИИ И ЭТАПЫ РАЗРАБОТКИ

Подготовительный этап разработки (Ноябрь 2022)

- Постановка задачи проекта;
- Сбор исходных теоретических материалов;
- Обоснование возможности решения поставленной задачи;
- Определение структуры входных и выходных данных;
- Предварительный выбор методов решения задач;
- Определение набора требований к программе;
- Определение требований к техническим средствам, использующим программу;
- Определение стадий и сроков разработки программы и ее документации;
- Согласование и утверждение технического задания.

Основной этап разработки (Декабрь 2022 – Март 2023)

- Выбор использующихся в проекте абстракций и типов данных;
- Выбор обобщенного подхода к описанию произвольной ISA тестируемой системы;
- Выбор обобщенного подхода к описанию частично заданных и зафиксированных тестов для произвольной тестируемой системы;
- Выбор обобщенного подхода к генерации случайных входных данных для инструкций в рамках тестовых последовательностей;
- Создание и получения необходимых для функционирования программы файлов с учетом разработанного алгоритма;
- Проведение испытаний разработанного проектного решения;
- Корректировка проектного решения в соответствии с полученными в процессе испытаний результатами.

Заключительный этап разработки (Апрель 2023 – Май 2023)

- Разработка программных документов в соответствии с наложенными на них требованиями;
- Подготовка презентации проекта к защите перед предметной комиссией НИУ ВШЭ;
- Защита проекта перед предметной комиссией НИУ ВШЭ.

Изм.	Лист	№ док.	Подп.	Дата
RU.17701729.04.04-01 ТЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

8. ПОРЯДОК КОНТРОЛЯ И ПРИЕМКИ

Виды испытаний

Производится проверка успешного и корректного выполнения программным решением под реализацию по умолчанию (ISA RISC-V RV32I) поставленных задач (функциональное тестирование программы). Проверка осуществляется в соответствии с документом «Прототип онлайн-генератора тестовых программ для микропроцессоров». Программа и методика испытаний (ГОСТ 19.301-79), устанавливающим:

- Перечень функций программы, подлежащих проверке, а также набор требований, которым должны следовать тестируемые функции (со ссылкой на пункт 4.1.1 «Состав выполняемых функций» настоящего технического задания);
- Перечень необходимой документации и требования к ней;
- Методы испытаний;
- Технические средства и порядок проведения испытаний.

В частности, среди методик испытания рассматривается ручное отслеживание сопроводительных и отладочных сообщений, а также исполняемых в ходе работы генератора инструкций с целью выявления расхождений с ожидаемыми результатами работы программы.

Общие требования к приемке работы

Прием программы будет осуществлен при успешном и корректном функционировании программы в соответствии с пунктом 4.1.1 «Состав выполняемых функций» при определенных входных данных (пункт 4.1.2 «Организация входных данных»), предоставляя определенные выходные данные (пункт 4.1.3 «Организация выходных данных»), а также при предоставлении полного набора технической документации по программе (пункт 5.1 «Состав программной документации») в соответствии с указанными к ним требованиями (пункт 5.2 «Специальные требования к программной документации»).

Изм.	Лист	№ док.	Подп.	Дата
RU.17701729.04.04-01 ТЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

9. СПИСОК ИСПОЛЬЗОВАВШИХСЯ ИСТОЧНИКОВ

- 1) Andrew Waterman, Krste Asanovic. The RISC-V Instruction Set Manual [Статья в Интернет]
Режим доступа: <https://riscv.org/wp-content/uploads/2017/05/riscv-spec-v2.2.pdf> , свободный.
(дата обращения: 14.11.2022)
- 2) CMake cross-platform build-process managing system. [Интернет-ресурс]
Режим доступа: <https://cmake.org/> , свободный. (дата обращения: 14.12.2022)
- 3) IBM Threadmill Post-Silicon Exerciser [Статья в Интернет]
Режим доступа: https://researcher.watson.ibm.com/researcher/view_group.php?id=1347 , свободный.
(дата обращения: 13.11.2022)
- 4) MicroTESK configurable test program generator [Статья в Интернет]
Режим доступа: <http://www.microtesk.org/> , свободный. (дата обращения: 13.11.2022)
- 5) Random instruction generator for RISC-V processor verification [Интернет-ресурс] Режим доступа: <https://github.com/chipsalliance/riscv-dv> , свободный (дата обращения: 09.01.2023)
- 6) А.С. Камкин, А.М. Коцыняк, С.А. Смолов, А.Д. Татарников, М.М. Чупилко, А.А. Сортов.
Средства функциональной верификации микропроцессоров [Статья в Интернет]
Режим доступа: Сборник трудов ИСП РАН, <https://ispranproceedings.elpub.ru/jour/article/view/771>, свободный. (дата обращения: 13.10.2022)

Изм.	Лист	№ док.	Подп.	Дата
RU.17701729.04.04-01 ТЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

10. ПРИЛОЖЕНИЯ**Приложение 1****Словарь использующихся терминов и понятий**

Таблица 1 – словарь использующихся терминов и понятий

Термин	Значение
Функциональная верификация	Процесс проверки соответствия разрабатываемой RTL-модели микропроцессора документам, описывающих требования к ней.
Динамическое (функциональное) тестирование	Способ проведения верификации, подразумевающий проверку соответствия реализации требованиям путем исполнения тестов на целевой (тестируемой) системе.
Оффлайн-тестирование	Процесс генерации набора тестовых ситуаций без использования системы, подлежащей тестированию.
Онлайн-тестирование	Процесс генерации набора тестовых ситуаций и их непосредственного исполнения на системе, подлежащей тестированию.
Микропроцессорная архитектура (Instruction Set Architecture, ISA)	Совокупность правил и свойств, по которым происходят вычисления на уровне микропроцессора. ISA регламентирует использование регистров микропроцессора, а также существующие типы данных, модель памяти и другие аспекты.
Bare-metal система	Комплекс аппаратных средств без выстроенного программного окружения, ответственного за контроль используемой памяти, планируемых задач и других взаимодействий в рамках системы и необходимых для этого ресурсов – операционной системы.

Изм.	Лист	№ док.	Подп.	Дата
RU.17701729.04.04-01 ТЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата