**Министерство образования и науки Российской Федерации**

**федеральное государственное автономное образовательное учреждение высшего образования**

**“САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ**

**УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ,**

**МЕХАНИКИ И ОПТИКИ”**

Факультет Программной инженерии и компьютерной техники

Образовательная программа Системное и прикладное программное обеспечение

Направление подготовки (специальность) Программная инженерия (09.03.04)

О Т Ч Е Т

об учебной практике

Тема задания: Исследование и моделирование программной модели аппаратной вычислительной архитектуры

Обучающийся Полещук Ф. А., гр. P3212

Руководитель практики от профильной организации:

Руководитель практики от университета: Логинов И. П., ассистент

Практика пройдена с оценкой \_\_\_\_

Подписи членов комиссии:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_ *\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

(подпись)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_ *\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

(подпись)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_ *\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

(подпись)

Дата \_\_\_\_

Санкт-Петербург

2019

1. Постановка задачи

В рамках поставленной задачи было необходимо описать любую целевую микропроцессорную архитектуру в нотации описания архитектур, предоставленной руководителем практики. По жребию из шести архитектур была выбрана архитектура MIPS.

Информация об архитектуре была взята из официальных источников [1] и [2].

1. Об архитектуре MIPS

Архитектура MIPS была разработана в Стэнфордском университете в 1981 году; первая аппаратная реализация её вышла в 1984. К настоящему времени MIPS развивается в двух ветвях – 32-битной MIPS32 и 64-битной MIPS64 ─, причём 64-битная ветвь является надмножеством 32-битной. Последняя версия архитектуры ─ MIPS32/64 Release 6 ─ вышла в 2014 году; эта версия и была реализована в ходе работы и в дальнейшем изложении под MIPS будет подразумеваться именно MIPS Release 6 с указанием разрядности когда это необходимо.

Вне зависимости от разрядности, процессоры архитектуры MIPS содержат 32 регистра общего назначения (РОН), однако регистр r0 используется как источник нулевого значения; запись в него игнорируется. MIPS поддерживает три сопроцессора (COP0-2), из которых COP0 используется для обеспечения прерываний, исключений и межпроцессорного взаимодействия, COP1 используется как FPU (математический сопроцессор), а реализация COP2 оставлена на усмотрение поставщика реализации.

MIPS не резервирует по умолчанию какие-либо области памяти для каких-либо целей. Максимальный объём адресуемой памяти зависит от разрядности архитектуры (4 ГиБ для MIPS32 и 16 ЭиБ для MIPS64): каких-либо ограничений на объём памяти кроме разрядности не предусмотрено.

Стеки организуются в адресном пространстве памяти путём хранения значения стека в каком-либо регистре. MIPS не предусматривает резервирования регистров под какие-либо цели в отличие, например, от Intel 64: по общепринятому соглашению для указателя на вершину стека используется регистр $r29, а для указателя на место в стеке (например, на нижнюю границу стекового кадра) ─ $r30.

Поддерживается виртуальная память и кэши: для них выделен отдельный набор инструкций.

Система команд MIPS Release 6 насчитывает более 120 инструкций. Размер инструкции MIPS – вне зависимости от разрядности – составляет 32 бита, из которых первые шесть – всегда код операции. Сами инструкции разделяются на три класса: R (регистровые), I (с непосредственным операндом) и J (перехода), ─ однако в ходе выполнения работы было выяснено, что такая классификация не отражает кодировку многих инструкций.

MIPS является архитектурой load/store: присутствует чёткое разделение между инструкциями работы с памятью (load/store) и остальными инструкциями. Все операции выполняются над операндами в регистрах. Кроме того, до MIPS Release 6 не было инструкций работы с отдельными битами или с невыровненными по границам слов данными. Вне зависимости от разрядности присутствуют инструкции для работы с байтами, полусловами в 2 байта и словами в 4 байта. MIPS64 добавляет работу с двойными словами в 8 байт.

Сами инструкции делятся на:

* Инструкции ввода-вывода;
* Инструкции АЛУ;
* Инструкции сопроцессоров (в т.ч. перемещения данных из CPU в COPx и обратно);
* Инструкции работы с кэшем;
* Инструкции работы с битовыми данными (например, выравнивание данных в операнде);
* Инструкции исключений.

1. О нотации описания архитектур и техническом обеспечении практики

Нотация описывает архитектуру как набор следующих элементов:

* Регистров;
* Представлений регистров: именованных отображений частей регистров или нескольких регистров одновременно;
* Модели памяти;
* Инструкций и их мнемоник.

Инструкции в нотации описываются как набор последовательных битовых полей трёх типов: непосредственных, с предварительно описанной кодировкой и альтернатив. Такое описание позволяет описать большую часть кодировок инструкций, но в некоторых случаях его становится недостаточно. Один из таких случаев будет описан ниже.

Техническое обеспечение практики включало в себя консольное приложение-клиент к удалённой системе, позволявшей проводить валидацию описаний архитектур, ассемблирование и дизассемблирование программ и несколько вспомогательных процедур наподобие генерации синтаксического дерева для описания архитектуры.

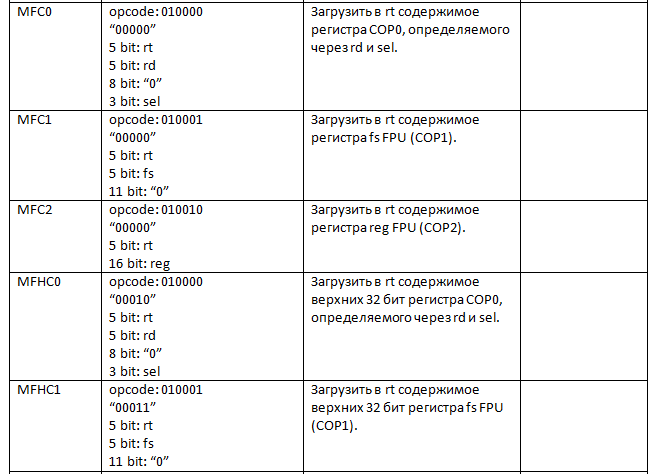
1. Выполнение

Выполнение работы проводилось по следующему плану:

1. Поиск, изучение и анализ документации. Был использован официальный источник «MIPS Architecture For Programmers», из которого были рассмотрены тома II и II-A, описывающие необходимые с точки зрения реализации детали;
2. Создание конспекта архитектуры. Было принято решение создать один конспект на MIPS32 и MIPS64, вынеся все дополнительные инструкции последней в отдельный раздел конспекта;
3. Реализация архитектуры в данной нотации;
4. Тестирование путём ассемблирования и дизассемблирования;
5. Оформление отчёта о проделанной работе.

В ходе реализации архитектуры потребовалось решить проблему, связанную с кодированием инструкций перехода, добавленных в MIPS Release 6. Проблема была связана с тем, что шесть новых пар инструкций перехода кодировались идентично, различаясь лишь значением поля кодирования регистра. Такое кодирование инструкций является нестандартным ходом и техническое обеспечение практики не позволило провести валидацию описания этих инструкций. Попытки разрешить эту проблему путём описания альтернативы кодирования регистрового поля для парных случаев окончились неудачей. После консультации с руководителями практики был сделан вывод, что описание такого рода инструкций невозможно.

Часть конспекта архитектуры приведена ниже на рисунке 1, часть описания архитектуры – в листинге 1, часть теста – в листинге 2.



*Рисунок 1. Часть конспекта архитектуры*

*Листинг 1. Часть описания архитектуры:*

instruction i\_sll

= {

000000,

00000,

reg as rt, reg as rd, imm5 as sa,

000000

}{};

instruction i\_sllv

= {

000000,

reg as rs, reg as rt, reg as rd,

00000,

000100

}{};

instruction i\_slt

= {

000000,

reg as rs, reg as rt, reg as rd,

00000,

101010

}{};

instruction i\_slti

= {

001010,

reg as rs, reg as rt, imm16 as value

}{};

*Листинг 2. Часть теста описания архитектуры:*

lb $r1, 0x0($r9)

lh $r2, 0x1($r10)

lw $r3, 0x2($r11)

lbu $r7, 0x2($r11)

lhu $r8, 0x3($r12)

lb $r17, 0x7($r9)

lh $r18, 0x8($r10)

lw $r19, 0x0($r11)

lw $r20, 0x4($r12)

add $r13, $r1, $r11

sub $r14, $r12, $r2

mul $r15, $r3, $r4

muh $r16, $r5, $r6

mulu $r17, $r7, $r8

muhu $r18, $r9, $r10

sb $r1, 0x1($r9)

sh $r2, 0x1($r10)

sw $r3, 0x1($r11)

sb $r17, 0x1($r9)

sh $r18, 0x1($r10)

sw $r19, 0x1($r11)

sw $r20, 0x1($r12)

AND $r1, $r3, $r5

OR $r7, $r9, $r11

XOR $r2, $r4, $r6

rotrv $r8, $r10, $r12

seleqz $r1, $r3, $r5

selnez $r7, $r9, $r11

slt $r2, $r4, $r6

sltu $r8, $r10, $r12

sllv $r1, $r3, $r5

srlv $r7, $r9, $r11

srav $r2, $r4, $r6

auipc $r1, 0x123

Описание тестировалось на образцах исходного текста суммарным объёмом 180 LoC (строк кода; в этом случае – инструкций). Тестирование показало корректность описания.

1. Результаты и выводы

В ходе выполнения работы была изучена архитектура MIPS, составлен её конспект и описание в нотации, данной руководителем практики; было проведено успешное тестирование описания. Были изучены официальные описания архитектуры MIPS.

По результатам работы в нотации и её техническом обеспечении были найдены следующие недостатки:

* Недостаточное покрытие реализации документацией;
* Невозможность или сложность описания некоторых форматов инструкций. Например, невозможно описать в данной нотации пару инструкций, различающихся только способом кодирования регистровых полей, как в примере выше;
* Неудобство реализации множества однотипных инструкций. Было бы желательно наличие возможности создавать форматы инструкций подобно форматам мнемоник;
* В реализации команда ассемблирования принимает в качестве входных параметров только один файл с описанием, что усложняет разделение реализации архитектуры на файлы. Если указано несколько одинаковых параметров, то берётся только последний, что не является решением проблемы.

1. Список литературы
2. MIPS32® Architecture For Programmers. Vol. II-A [Электронный ресурс] // MIPS32 Architecture. URL: https://s3-eu-west-1.amazonaws.com/downloads-mips/documents/MD00086-2B-MIPS32BIS-AFP-6.06.pdf (дата обращения: 28.05.19)
3. MIPS64® Architecture For Programmers. Vol. II-A [Электронный ресурс] // MIPS64 Architecture. URL: https://s3-eu-west-1.amazonaws.com/downloads-mips/documents/MD00087-2B-MIPS64BIS-AFP-6.06.pdf (дата обращения: 28.05.19)