Оглавление

Этап 1. Исследование архитектуры POWER

Ознакомление с историей развития архитектуры

Согласно варианту, архитектурой, подлежащей рассмотрению, является архитектура POWER разработчика IBM. Расшифровка аббревиатуры – Performance Optimization With Enhanced RISC (reduced instruction set computer).

Первая поставленная передо мной задача – познакомиться с историей версий выбранной архитектуры. Для этого были изучены статьи «POWER» [1] и «Power ISA»[2]. Дерево версий представлено на Рис. 1 История развития архитектуры POWER в Приложении.

Ознакомление с выбранной моделью архитектуры

Проанализировав актуальность версий, для дальнейшего рассмотрения была выбрана модель POWER ISA (version 3.0). Для ознакомления с ней была изучена документация. [3] Конкретно были проанализированы следующие разделы:

- Book I: Power ISA User Instruction Set Architecture
  - Chapter 1. Introduction. (Общее представление о программной модели архитектуры, включая модель памяти, информацию о регистрах).

При описании модели архитектуры было отмечено наличие различных устройств для работы с различными типами данных, таких как:

- calar
  - fixed-point
  - floating-point
- vector
  - fixed-point
  - floating-point

Модель обработки типов данных представлена на Рис. 1

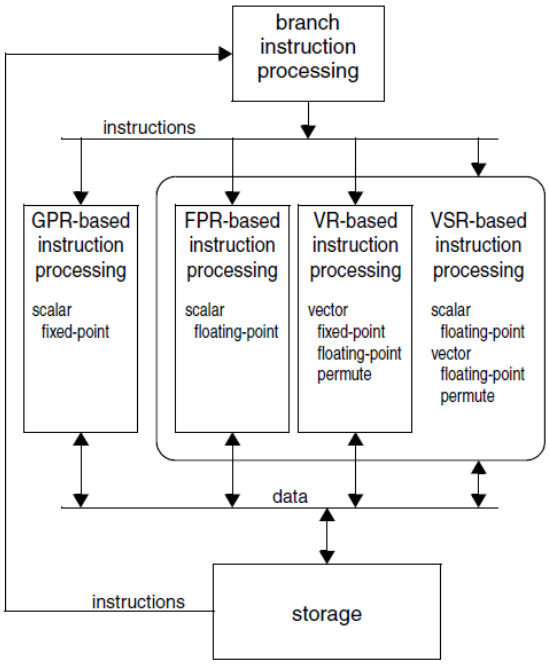

Рис. 1 Модель логической обработки

Каждое устройство имеет собственный набор регистров для собственного функционирования.

Устройство, отвечающее за команды перехода, имеет:

- 32-битный регистр для хранения результата определенных операций. Этот результат, используется командами условного перехода (CR)
- 64-битный регистр для хранения адреса возврата (LR)
- 64-битный регистр с количеством циклических переходов (CTR)
- 64-битный регистр для хранения адреса назначения (TAR)

Устройство, отвечающее за работу с числами с фиксированной запятой, имеет:

- 32 регистра (64-битные) общего назначения (GPR0-GPR31)

- 64-битный регистр, отвечающий за исключение полученные в результате при работе с числами с фиксированной запятой

Устройство, отвечающее за работу с числами с плавающей запятой, имеет:

- 32 регистра (64-битные) общего назначения (FPR0-FPR31)

Для модели характерно единое адресное пространство.

Следующим шагом изучения модели архитектуры является знакомство с набором инструкций. Для ознакомления была использована документация - Power ISA™ Version 3.0. А именно следующие разделы:

- Book I: Power ISA User Instruction Set Architecture
  - Chapter 2. Branch Facility (Описание инструкций и регистров для работы с командами перехода)
  - Chapter 3. Fixed-Point Facility (Описание инструкций и регистров для работы с числами с фиксированной запятой)
  - Chapter 4. Floating-Point Facility (Описание инструкций и регистров для работы с числами с плавающей запятой)

По окончанию изучения материала был структурирован набор инструкций и составлена схема, изображенная на Таблицах 1-3 в Приложении.

Этап 2. Исследование инструментария для создания модели прикладной архитектуры процессора

Знакомство с документацией, представленной руководителем практики

Для реализации модели необходимы для изучения:

- Правила написания нотации
- Описание структуры файлов .asm
- Описание структуры файлов .bin

Знакомство с тестовым окружением, предоставленным преподавателем

Для анализа работы модели, которую необходимо разработать, руководителем было предоставлено тестовое окружение и поставлена задача познакомиться с возможными опциями и функционалом в целом.

Конкретно нас интересовали возможности проверки кода на валидность, ассемблирование .asm файла и дисассемблирование .bin файла.

Этап 3. Реализация модели

Часть 1. Описание регистров

```
storage CR[32];
view cr0 = CR[0..3];
view cr1 = CR[4..7];
view cr2 = CR[8..11];
view cr3 = CR[12..15];
view cr4 = CR[16..19];
view cr5 = CR[20..23];
view cr6 = CR[24..27];
view cr7 = CR[28..31];

view lt = CR[0];
view gt = CR[1];
view eq = CR[2];
view so = CR[3];
view un = CR[3];

storage LR[64];
storage CTR[64];

storage TAR[64];
view TAR_EA = TAR[0..61];
```

```
storage XER[64];
view SO = XER[32];
view OV = XER[33];
view CA = XER[34];
view OV32 = XER[44];
view CA32 = XER[45];

storage VRSAVE[32];

storage LMRR[64];
view Base_EA = LMRR[0..38];
view Size = LMRR[60..63];

storage LMSER[64];
```

Листинг 1 Пример конфигурации регистров модели и их представлений

Часть 2. Описание блоков памяти

Для данной архитектуры характерно единое адресное пространство.

```
| memory: range ram[0x00000000..0xFFFFFFFF]
```

Часть 3. Описание набора инструкций

```
encode imm7 field = immediate[7];
encode imm5 field = immediate[5];
encode imm3 field = immediate[3];
encode imm2 field = immediate[2];
encode imm1 field = immediate[1];
encode CR_fields field = register {
        cr0 = 000,
        cr1 = 001,
        cr2 = 010,
        cr3 = 011,
        cr4 = 100,
        cr5 = 101,
        cr6 = 110,
        cr7 = 111
};
encode CR_field field = register {
        lt = 00,
        gt = 01,
        eq = 10,
        so = 11,
        un = 11
};
```

Листинг 2 Пример написания битовых паттернов

```
instruction bclr_1 = {010 011, imm5 as BO, imm5 as BI, rsvd3.bits0, imm2 as BH, 00001 00000, 0};
instruction bclr_2 = {010 011, imm5 as BO, imm5 as BI, rsvd3.bits0, 00,          00001 00000, 0};

instruction lbz    = {100 010, imm5 as RT, imm5 as RA, imm16 as D};
instruction lbzu   = {100 011, imm5 as RT, imm5 as RA, imm16 as D};
instruction lhz    = {101 000, imm5 as RT, imm5 as RA, imm16 as D};
instruction lhzu   = {101 001, imm5 as RT, imm5 as RA, imm16 as D};
```

Листинг 3 Пример кодирования полей в составе опкода

Часть 4. Описание мнемоник команд

```
format I is "{1}";
format B is "{1},{2},{3}";
format XL_2 is "{1},{2}";
format XL_3 is "{1},{2},{3}";
```

Листинг 4 Пример описания форматных строк

```
mnemonic b for b (LI) I;
mnemonic ba for ba (LI) I;
mnemonic bl for bl (LI) I;
mnemonic bla for bla (LI) I;

mnemonic bclr    for bclr_1 (BO, BI, BH) XL_3;
mnemonic bclr    for bclr_2 (BO, BI) XL_2;
```

Листинг 5 Пример описания мнемоник команд

Результаты

Тестирование реализованной модели: проверка валидности кода

```
Architecture: POWER v3.0
TaskID: 125654d2-cba5-4e9c-826b-5117ee981086

RESULTS:
Task creation time: 27.05.2019 15:21:24
Task state: Finished
Task finish time: 27.05.2019 15:21:24
Here is task output interpreted with UTF8:
Architecture ' POWER v3.0' seems to be valid for now.
```

Листинг 6 Результат проверки кода на валидность

Тестирование реализованной модели: ассемблирование

```
b 11184810
bc 16, 0, 0x2AAA
bc 8,2,0x2AAA
bc 8,22,0x2AAA
bc 4,27,0x2AAA
bcl 4,27,0x2AAA
bc 4,2,0x2AAA
bc 4,14,0x2AAA
bcla 12,17,0x2AAA
bcctrl 12,17,1
creqv 25,25,25
crxor 3,3,3
crxor 15,15,15
crnor 2,2,2
crnor 22,18,18
```

Листинг 7 Пример содержимого .asm файла

```
Architecture: POWER v3.0
TaskID: 4f28f1dc-cf7f-4ea1-958f-06a22c91ceff

RESULTS:
Task creation time: 27.05.2019 15:21:38
Task state: Finished
Task finish time: 27.05.2019 15:21:38
Here is task output interpreted with UTF8:
Architecture ' POWER v3.0' seems to be valid for now.
Assembling accomplished.
Binary module written.
```

Листинг 8 Результат ассемблирования файла

Тестирование реализованной модели: дисассемблирование

```
RESULTS:
Task creation time: 27.05.2019 15:21:52
Task state: Finished
Task finish time: 27.05.2019 15:21:52
Here is task output interpreted with UTF8:
Binary module disassembled.
```

Листинг 9 Результат дисассемблирования файла

Анализ результатов тестирования

Итогом выполнения практического задания является описание модели архитектуры (файл .target.pdsl) Проведение тестирования подтверждает корректность модели. Операции ассемблирования и дисассемблирования оканчиваются успехом. Исходный файл совпадает с результатом дисассемблирования. Модель может быть применима в других проектах.

Список литературы

1. Архитектура POWER [Электронный ресурс] URL – https://ru.wikipedia.org/wiki/POWER#POWER7
2. POWER ISA [Электронный ресурс] URL – https://en.wikipedia.org/wiki/Power_ISA#Power_ISA_v.3.0
3. Книга Power ISA™ Version 3.0