

Моделирование OpenRISC 1000

Григорий Речистов
grigory.rechistov@intel.com

26-28 августа 2014 г.



1 Обзор or1k

2 Ход работы

3 Литература

4 Конец



Что такое OpenRISC 1000

- Модульная архитектура цифровой системы с поддержкой сообщества OpenCores.
- Спецификация центрального процессора/SoC.
- Реализована в симуляторах, VHDL, FPGA и кремнии.
- Поддерживается Linux, Glibc и др.



Почему OpenRISC 1000 для симуляции



Почему OpenRISC 1000 для симуляции

- MIPS всем надоел.



Почему OpenRISC 1000 для симуляции

- MIPS всем надоел.



Почему OpenRISC 1000 для симуляции

- MIPS всем надоел.
- Модульная структура — много необязательных частей.
- Простая архитектура.

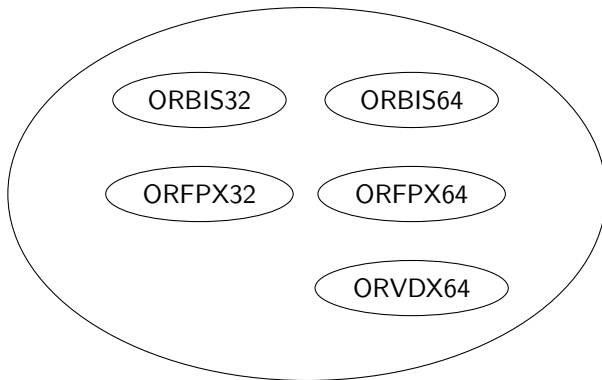


Почему OpenRISC 1000 для симуляции

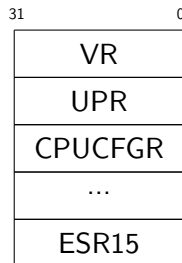
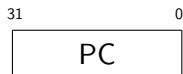
- ~~MIPS~~ всем надоел.
- Модульная структура — много необязательных частей.
- Простая архитектура.
- Но не *слишком* простая.



ISA



Регистры



Что ещё есть

- Исключения.
- TLB
- MMU.
- Cache.
- PIC.
- Tick timer.
- PCU, Debug, PMU ...



Получение исходников модели

```
$ git clone user@host:/path wrkspc  
Cloning into 'wrkspc'...  
done.
```



Создание workspace

```
$ /opt/simics/simics-4.6/simics-4.6.112/bin/workspace-setup wrkspc  
Workspace created successfully  
$ cd wrkspc
```



Сборка

```
$ make
```



Юнит-тест

```
$ ./bin/test-runner  
...  
Ran 3 tests in 3 suites in 2.171517 seconds.  
All tests completed successfully.
```



Структура кода

```
$ ls -1 modules/or1k/  
Makefile  
or1k-cycle.c  
or1k-cycle.h  
or1k-exec.c  
or1k-exec.h  
or1k-frequency.c  
or1k-frequency.h  
or1k-memory.c  
or1k-memory.h  
or1k-queue.c  
or1k-queue.h  
or1k-step.c  
or1k-step.h  
or1k.c  
or1k.h  
commands.py  
event-queue-types.h  
event-queue.c  
event-queue.h
```



Атрибуты

```
simics> api-help SIM_register_typed_attribute
Help on API keyword "SIM_register_typed_attribute":

// defined in simics/base/conf-object.h
#include <simics/device-api.h>    // in C/C++
// always available in DML

int SIM_register_typed_attribute(conf_class_t * NOTNULL cls, const char * NOTNULL name,
get_attr_t get_attr, lang_void *user_data_get, set_attr_t set_attr, lang_void
    *user_data_set, attr_attr_t attr, const char *type, const char *idx_type, const char *desc);

// available in Python
```



Интерфейсы

```
simics> api-help SIM_register_interface
Help on API keyword "SIM_register_interface":

// defined in simics/base/conf-object.h
#include <simics/device-api.h>    // in C/C++
// always available in DML

int SIM_register_interface(conf_class_t * NOTNULL cls, const char * NOTNULL name,
const interface_t * NOTNULL iface);

// available in Python
```






Задание

1. Выбрать часть функциональности, требующую реализации.
2. Спланировать процесс реализации.
3. Реализовать в коде.
4. Добиться успешной сборки.
5. Написать юнит-тест.
6. Добиться успешного завершения юнит-теста.
7. GOTO 1.



Литература I

-  Основы программного моделирования ЭВМ. Учебное пособие / Г. Речистов, А. Иванов, П. Шишпор, Н. Щелкунов, Д. Гаврилов, В. Пентковский. — Издательство МФТИ, дек. 2012. — ISBN 978-5-7417-0469-1
-  Open Cores OpenRISC 1000 Architecture Manual. Architecture Version 1.0 Document Revision 0. — 2012. — URL: <http://opencores.org/websvn,filedetails?repname=openrisc&path=/openrisc/trunk/docs/openrisc-arch-1.0-rev0.pdf>
-  Processor Model Integration Guide 4.6 / Wind River. — 2014.
-  Simics Model Builder User Guide 4.6 / Wind River. — 2014.



Спасибо за внимание!

Слайды и материалы курса доступны по адресу
<http://bit.ly/1y1lZF1>

Замечание: все торговые марки и логотипы, использованные в данном материале, являются собственностью их владельцев. Представленная точка зрения отражает личное мнение автора.

