

Московский физико-технический институт (государственный университет)

Факультет радиотехники и кибернетики

Кафедра информатики и вычислительной техники

Выпускная квалификационная работа бакалавра

Разработка тестового окружения для верификации контроллеров периферийных интерфейсов

Студент: Рудаков Константин Андреевич, 913 группа

Научный руководитель: к.т.н. Волконский Владимир Юрьевич

Введение

При верификации южного моста КПИ-1 в отделе было реализовано тестовое окружение:

- облегчающее разработку тестов
- позволяющее запускать тесты на функциональной и RTL моделях, а также прототипе вычислительного комплекса (ВК)

На основе тестового окружения было разработано порядка тысячи тестов на языке C++:

- Тесты запускаются в отсутствие ОС
- Тесты предназначены для различных архитектур: Elbrus-S, Elbrus-2C+, Elbrus-4C, Elbrus-8C, Elbrus-1C+

Структура тестового окружения

- Скрипты компиляции и запуска:
 - Сборка библиотеки
 - Сборка RTL
 - Компиляция и запуск теста
- Библиотека
 - Платформозависимая часть
 - Инициализация ВК
 - Платформонезависимая часть
 - Работа с PCI
 - Работа с прерываниями
 - Инициализация устройств (КПИ, LAPIC, PCI/PCIe Bridge)

Постановка задачи

Требуется доработка тестового окружения

- Модернизация механизма конфигурирования ВК
- Реализация возможности параллельного запуска существующих тестов на разных ядрах
- Модификация скриптов сборки и запуска

при условии сохранения полной совместимости с существующими тестами

С использованием доработанной библиотеки требуется разработать тесты, направленные на верификацию PCI/PCI Express моста, клиентской частью которого является имитатор PCI-устройства.

Механизм конфигурирования ВК

Требования

Старая версия: конфигурация ВК задавалась с помощью макроопределений и не позволяла специфицировать все необходимые параметры.

Требования к новому механизму:

- Задание параметров конфигурации из теста
- Реализация параметров ВК по умолчанию
- Сохранение производительности

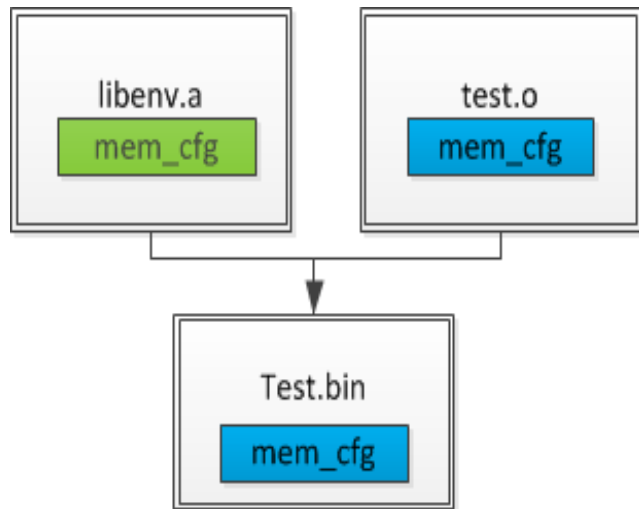
Механизм конфигурирования ВК

Реализация

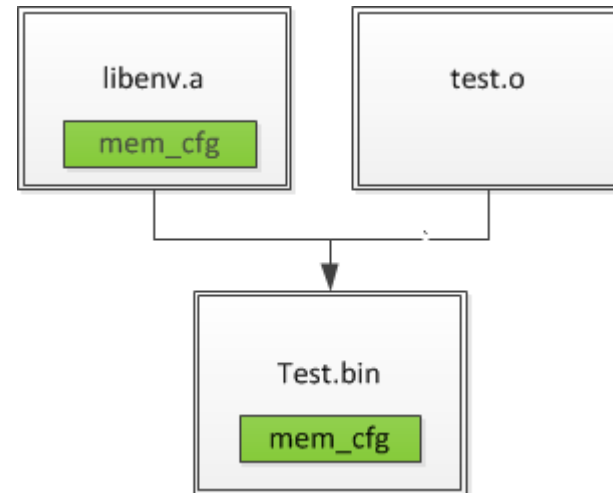
Для описания конфигурации используется глобальная константная структура.

```
extern const struct mem_cfg_t mem_cfg __attribute__((weak));
```

- `const` – быстроедействие (оптимизации времени компоновки)
- `weak` – значение по умолчанию



Определение параметров в тесте



Использование параметров по умолчанию

Механизм конфигурирования ВК

Примеры использования

Структура описания карты памяти

```
struct __mem_cfg_t
{
    uint32 mem_hi[4];
    uint32 mem_lo[4];
    uint32 mem_pci_m[4];
    uint32 mem_pci_io[4];
    uint32 mem_ioapic[4];
    uint32 mem_pci_mp_b[4];
    uint32 mem_pci_mp_e[4];
    uint32 mem_pci_cfg_b;
    uint32 mem_ioapic_int_b;
    uint32 mem_lapic_int_b;
    uint32 mem_sapic_int_b;
};
```

Используется при конфигурировании:

- Карты памяти (северного моста)
- Точек входа ядер
- Южного моста

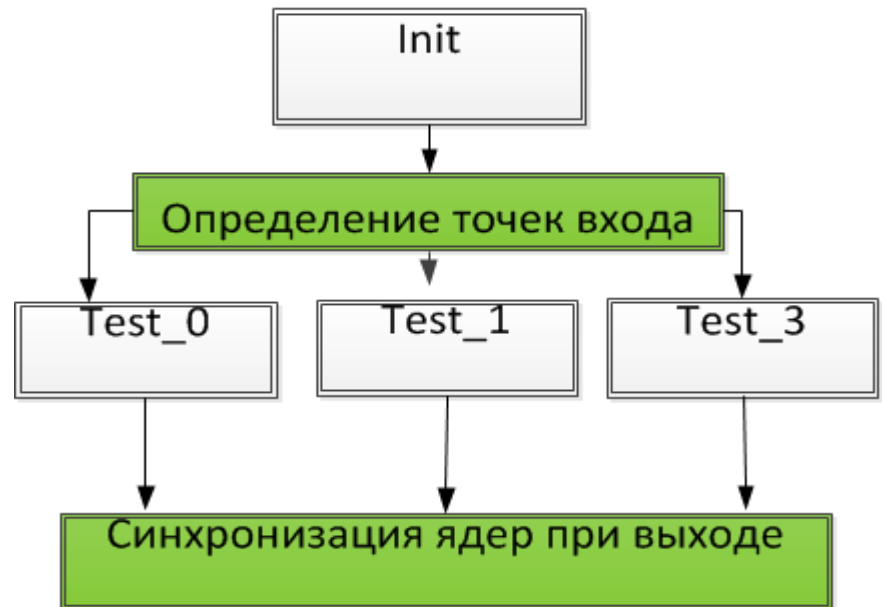
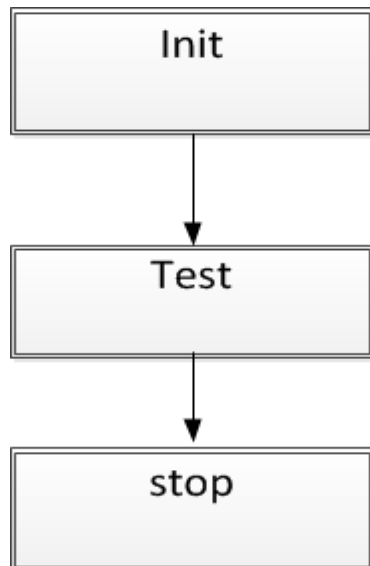
Параллельный запуск тестов

Актуальность

- Имеется база тестов для проверки периферийных контроллеров в одноядерном режиме
- Необходимо проверять одновременную работу нескольких контроллеров путем запуска нескольких тестов на разных ядрах

Реализация

- Разделение общих ресурсов (программный стек, стек связующей информации, пользовательский стек)
- Задание точек входа для различных ядер
- Синхронизация ядер при завершении тестов



Скрипты сборки и запуска

Старые скрипты

- Реализована компиляция/запуск тестового сценария
- Поддержка многопроцессорности (но не многоядерности)
- Реализованы на bash

Новые требования

- Поддержка параллельного запуска тестов
- Возможность запуска наборов тестовых сценариев
- Использование языка, более удобного для поддержки кода

Реализация

- Язык реализации - Python 2.7
- Поддержан механизм, обеспечивающий параллельный запуск тестов на разных ядрах
- Реализована возможность запуска набора тестовых сценариев
- Добавлена возможность запуска теста, состоящего из нескольких исходных файлов

Тесты для PCI/PCIe моста КПИ-2

Требования к тестам

Тесты используют имитатор PCI устройства в качестве клиента PCI/PCIe моста.

- Возможность работы как с PCI, так и PCIe мостами
- Поддержка всех режимов работы PCI имитатора:
 - DMA запись из буфера
 - DMA запись константы
 - DMA запись адресного кода
 - DMA запись обратного адресного кода
 - Вышеперечисленные режимы с инкрементом старшего и/или младшего двойного слова
 - DMA чтение

Тесты для PCI/PCIe моста

Этапы теста

- Инициализация южного моста
- Инициализация PCI/PCI Express моста
- Инициализация имитатора PCI устройства и проверка записи/чтения из его операционного пространства
- Подготовка операции DMA_read/DMA_write
- Начать операцию записи или чтения
- Дождаться окончания операции записи/чтения (используя прерывание или по опросу)
- Сравнить результаты с ожидаемыми

Результаты

Доработано тестовое окружение для верификации контроллеров периферийных интерфейсов:

- Реализован новый механизм конфигурирования ВК
- Реализована возможность параллельного запуска существующих тестов на разных ядрах
- Переработаны скрипты сборки и запуска
- Сохранена совместимость с существующими тестами
- Проведена ревизия кода, написана документация

Разработан и внедрён набор тестов для PCI/PCIe моста, обнаружены ошибки в имитаторе PCI устройства

Спасибо за внимание