

Московский физико-технический институт
(государственный университет)
Факультет радиотехники и кибернетики
Кафедра информатики и вычислительной техники

Автономная верификация программируемого контроллера прерываний многоядерного микропроцессора

Студент: Рослов Николай Александрович, группа 113

Руководитель: Стотланд Ирина Аркадьевна, к.т.н.

Постановка задачи

Цель: провести автономную верификацию контроллера прерываний многоядерного микропроцессора «Эльбрус-8С2».

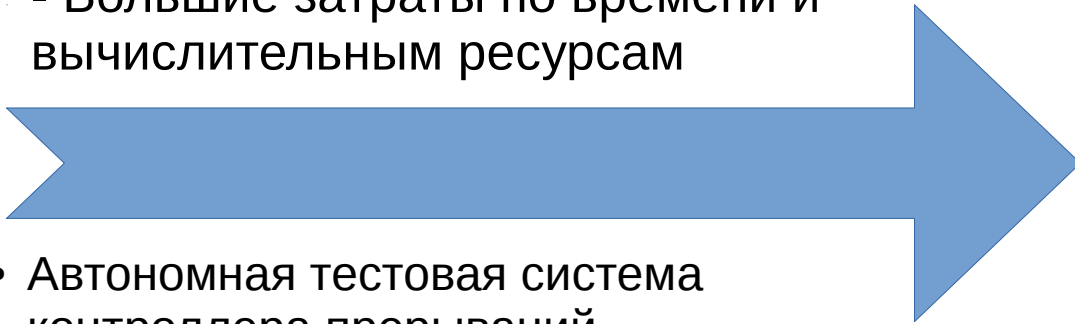
Задачи:

1. Разработка конфигурируемой тестовой системы.
2. Разработка набора тестов для RTL-модели контроллера.
3. Разработка системы регрессионного прогона тестов на имеющемся вычислительном кластере.

Разработка тестовой системы

Существующие решения

- Системные тесты:
 - Сложность достижения граничных ситуаций
 - Большие затраты по времени и вычислительным ресурсам
- Автономная тестовая система контроллера прерываний микропроцессора «Эльбрус-4С»:
 - Неконфигурируемая — трудоемкое повторное использование
 - Затруднено добавление новых тестов
 - Работа непосредственно с физическими интерфейсами устройства

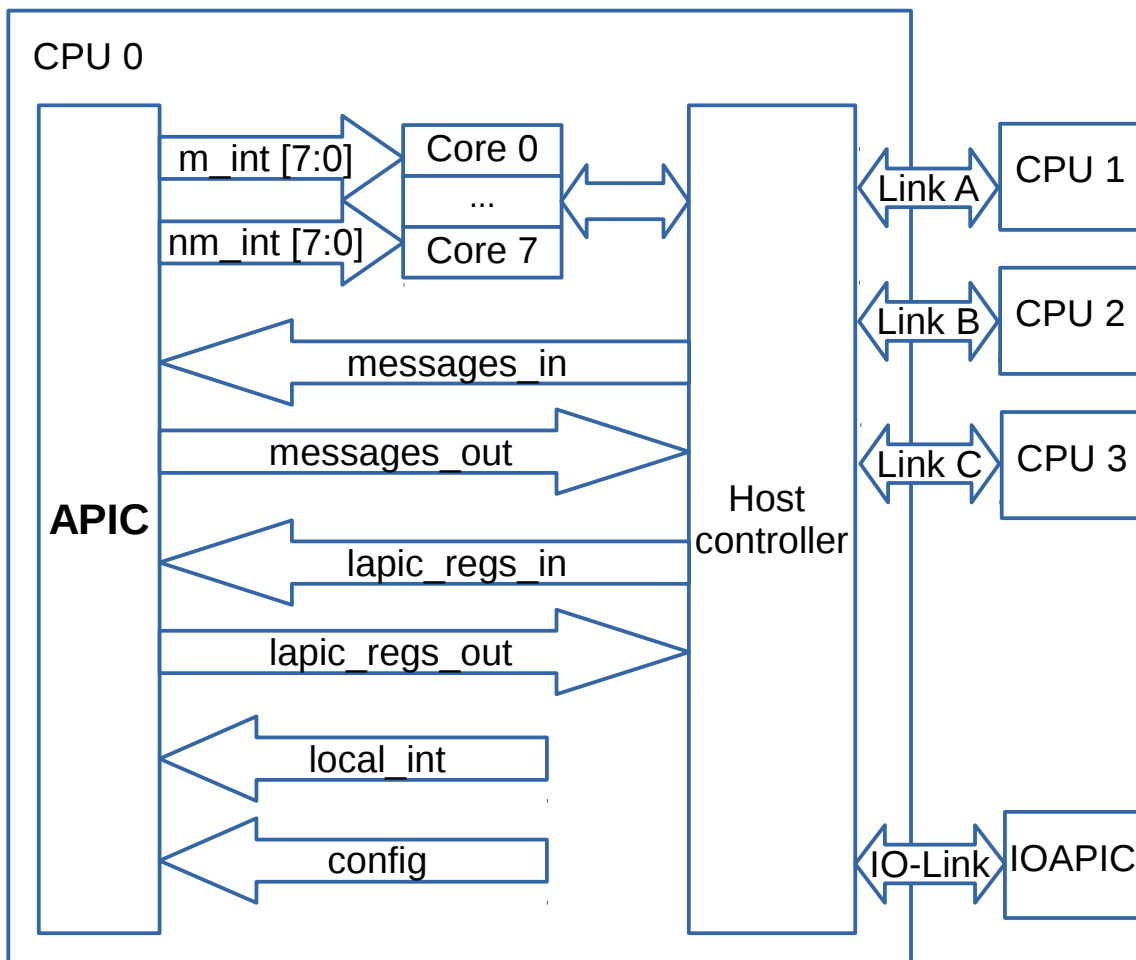


Выбранное решение

- Автономная тестовая система уровня транзакций (TLM)
 - + Логика работы не привязана к конкретной физической реализации
 - + Легкость повторного использования
- Реализация с использованием методологии UVM (библиотека на языке SystemVerilog)
- Модуль проверки на основе эталонной модели

Разработка тестовой системы

Характеристика тестируемого устройства



Типы транзакций

- Прерывания в ядро (m_int, nm_int)
- Сообщения о прерываниях (messages_in, messages_out)
- Запись в регистры (lapic_regs_in)
- Чтение из регистров (lapic_regs_out)
- Локальные прерывания (local_int)
- Задание конфигурации (config)

Направления передачи сообщений о прерываниях в многопроцессорной системе:

- В другие процессоры (CPU)
- В контроллеры внешних прерываний (IOAPIC)

Разработка тестовой системы

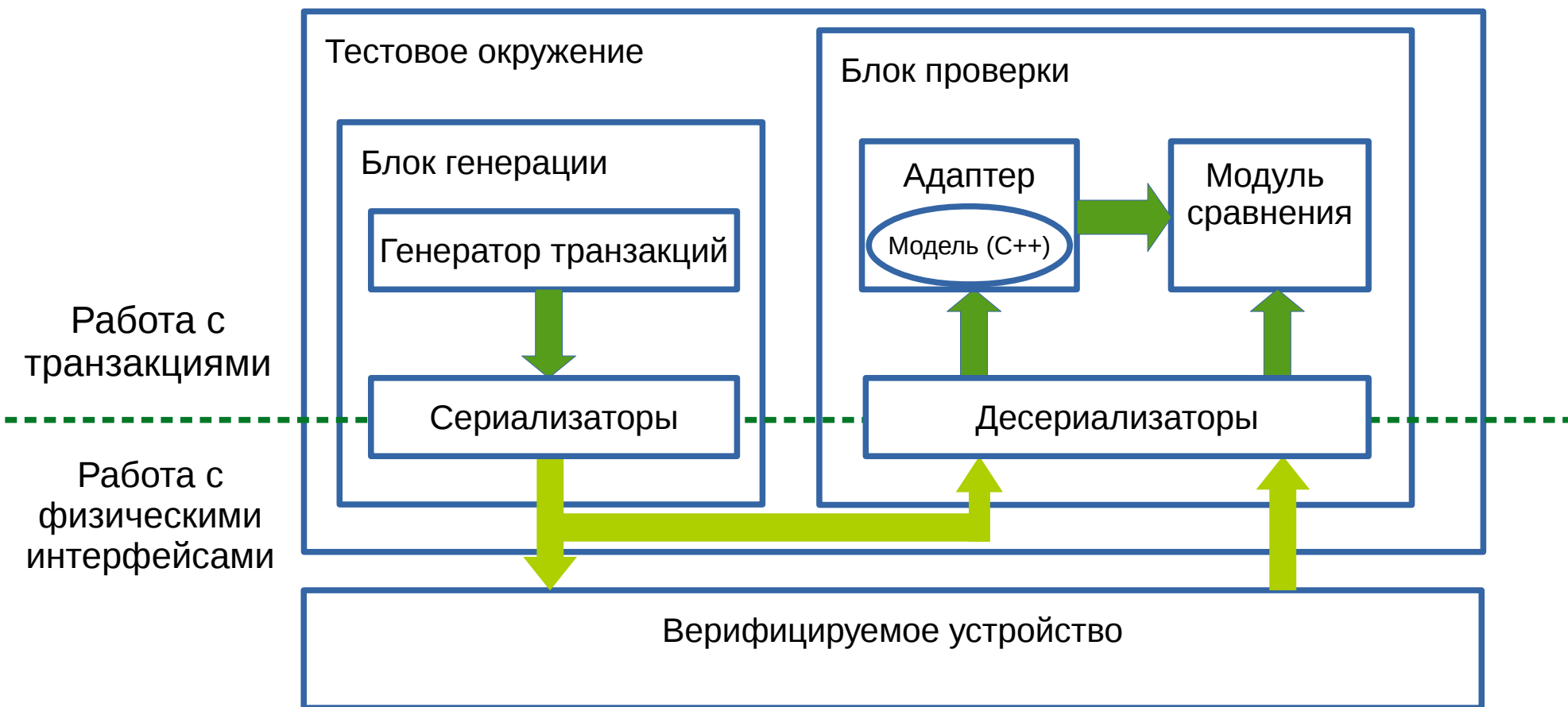
Выбор типа эталонной модели

Функциональная

- + работает на уровне транзакций
- + существует в составе системного симулятора
- + реализована другим инженером – избегаем дублирования ошибок
- не учитывает временные характеристики – не заданы в спецификации устройства
- реализована на C++ – требуется адаптер

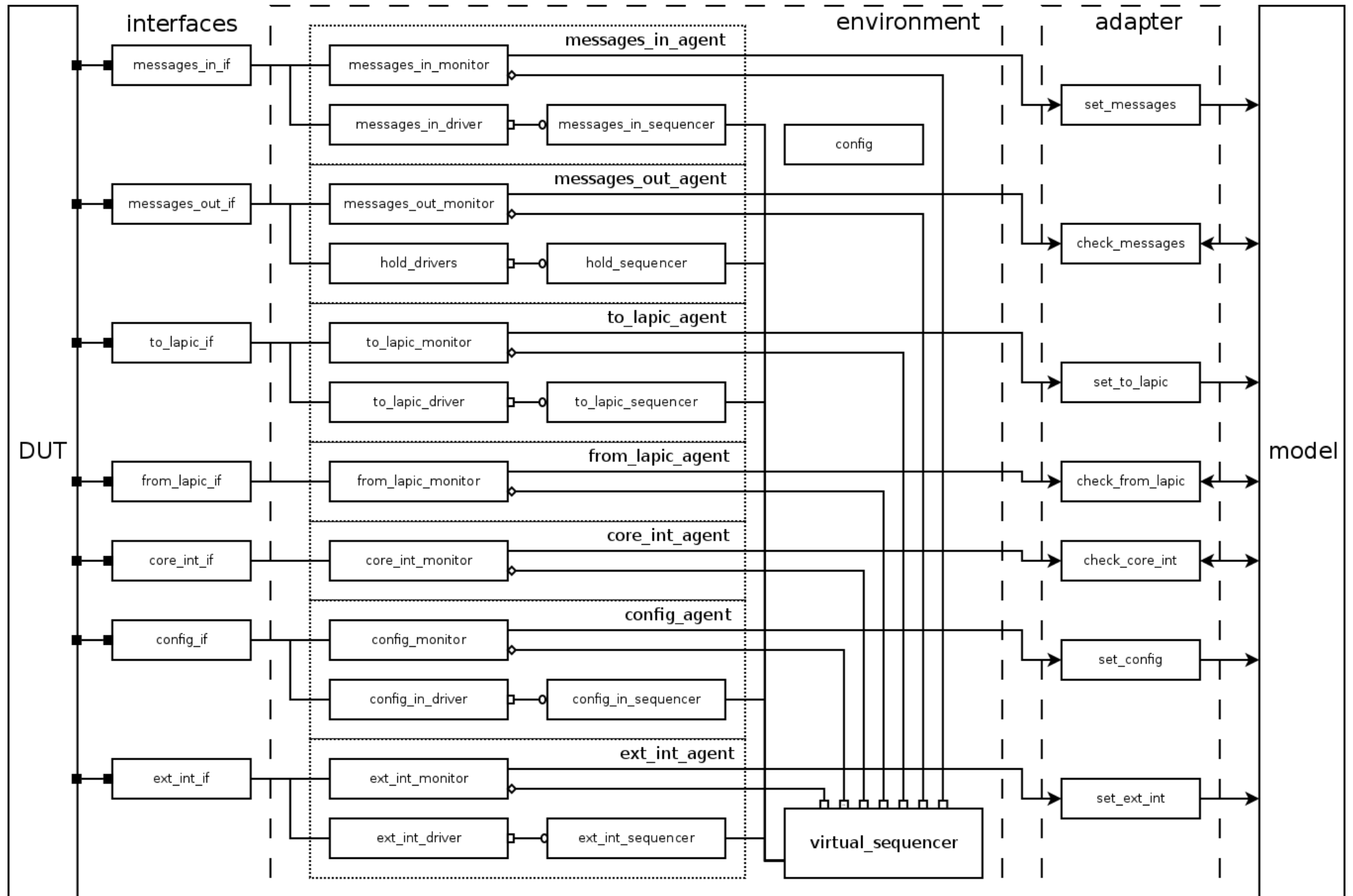
Разработка тестовой системы

Принцип построения тестовой системы



Разработка тестовой системы

Практическая реализация тестовой системы согласно методологии UVM



Тестовые сценарии

- **apic_base_test** – базовый тестовый сценарий:
 - содержит общую для всех тестов функциональность (вывод сообщения о успехе или неуспехе теста, установку таймаута и т. п.)
 - служит для проверки корректности сборки тестового окружения и RTL-модели
- **apic_read_reg_test** – корректность считывания начальных значений внутренних регистров локальных частей контроллера прерываний (LAPIC)
- **apic_id_test** – переназначение логических и физических идентификаторов ядер в контроллере:
 - корректность обновления соответствующих внутренних регистров
 - выдача сообщений контроллерам других процессоров системы
- **apic_message_test** – работа контроллера с сообщениями о прерываниях:
 - обработка сообщений от других процессоров и контроллера периферийных интерфейсов (IOAPIC)
 - выдача прерываний на основании сообщений
 - выдача ответных сообщений в КПИ и другие процессора

Тестовые сценарии

- **apic_timer_int_test** – работа программируемого таймера контроллера прерываний:
 - выдача прерывания по завершении счета
 - выдача сообщений
- **apic_timer_nm_int_test** – аналогичный **apic_timer_int_test**, но для немаскируемого таймера
- **apic_ext_int_test** – обработка прерываний, полученных от других устройств процессора:
 - выдача соответствующих прерываний в ядро
 - выдача сообщений о них в контроллеры других процессоров
- **apic_icr_int_test** – работа контроллера при вызове прерывания через регистр локального контроллера прерываний:
 - выдача прерываний в ядра
 - корректность обработки прерывания (вектор, приоритет и т. д.)
 - выдача сообщений о прерываниях в контроллеры других процессоров системы
- **apic_bc_icr_int_test** – аналогичный **apic_icr_int_test**, но работающий с широкопередаточными прерываниями:
 - работа очередей на выдачу прерываний на различных уровнях иерархии
 - корректность очистки очередей при получении квитанций об обработке прерывания
 - выдача сообщений о прерываниях

Система регрессионного прогона

Прежние решения:

- Скрипт для запуска на одной машине
 - занимает много времени => неудобен для использования при отладке тестов.
- Система «testrun» сектора системной верификации
 - иной интерфейс запуска тестов.

Предлагаемое решение:

Реализация системы прогона автономных тестов с использованием кластера в виде надстройки над SGE.

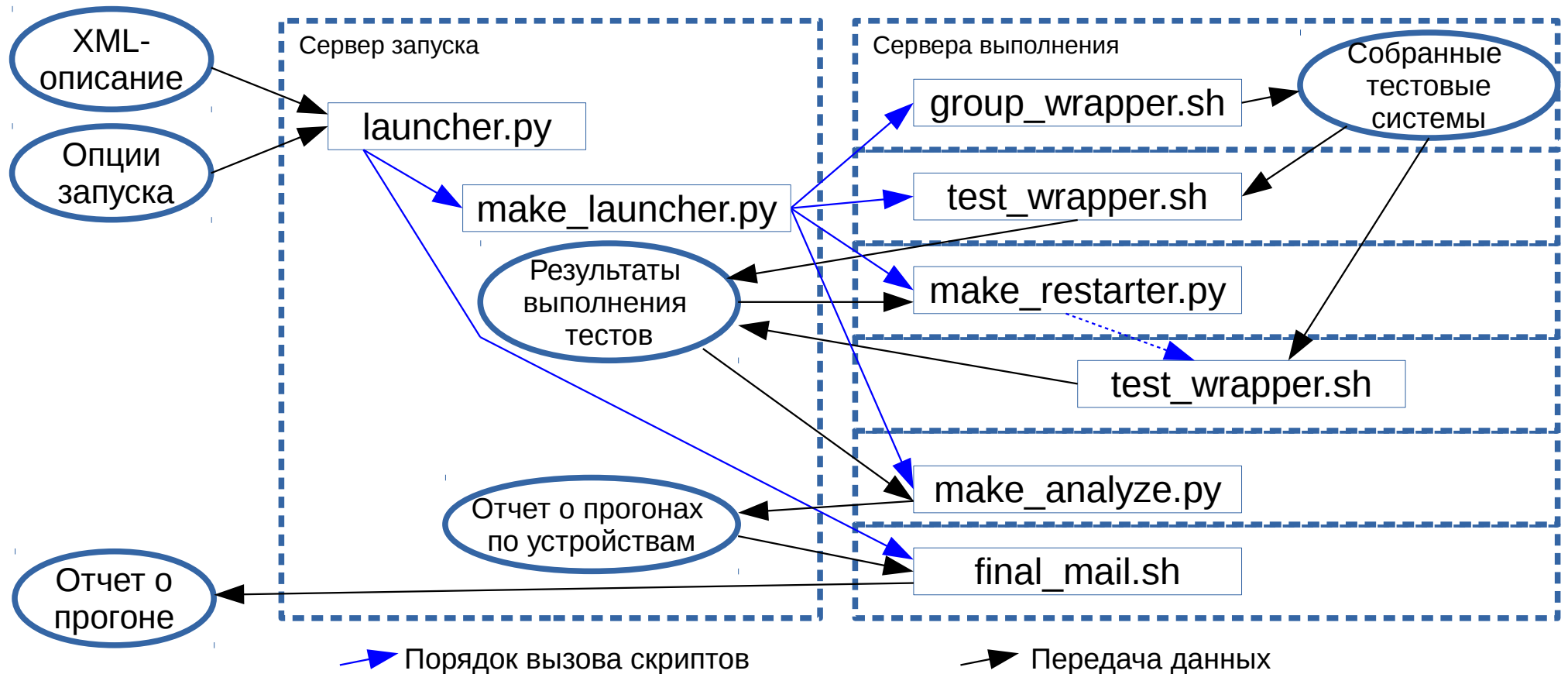
Система регрессионного прогона

Набор скриптов на языках python и bash + SGE:

- `launcher.py` – разбор опций запуска и списка устройств
- `make_launcher.py` – запуск сборки проекта и запуск массива тестов
- `make_restarter.py` – перезапуск тестов в случае возникновения ошибок
- `make_wrapper.sh` – вспомогательные действия до и после выполнения теста
- `make_analyzer.py` – анализ результатов выполнения теста и формирование отчета
- `make_xml_maker.py` – автоматизированное создание xml-описания задания

Система регрессионного прогона

Схема взаимодействия скриптов



- make_launcher.py запускается для каждого тестового окружения, участвующего в прогоне
- group_wrapper.sh запускается для каждой группы тестов, отличающихся параметрами сборки
- test_wrapper.sh и make_restarter.py запускается для каждого теста
- test_wrapper.sh запускается повторно только в случае ошибки при первом запуске

Система регрессионного прогона

Пример отправляемого пользователю отчета

StAlone run results for HC (project P9), launched at 16-03-23_15-05

Revisions: RTL: 5413; environment: 5415

Tests run: 4

Tests passed: 1

Testname	Seed	Host
hc_test_42	3069903937	irc-150

Tests failed: 3

Testname	Seed	Host
hc_test_42	3069903937	irc-150
hc_test_95	ERROR - no seed	irc-133
hc_test_99	ERROR - no out	irc-124

XML description of current launch:

/home/username/SVN/stalone/standalone/trunk/st_alone_run/devices/P9/HC/progon.xml

Report files: /net/irc-

111/export/data/verif/username/st_alone_rundir/P9/HC/16-03-23_15-05/

Logs of the failed tests: /net/irc-

111/export/data/verif/username/st_alone_rundir/P9/HC/16-03-23_15-05/failed_logs/

Time spent: 24 min.

Результаты работы

- Разработаны:
 - тестовая система для контроллера прерываний многоядерного микропроцессора на основе методологии UVM
 - система регрессионного прогона тестов на имеющемся в АО «МЦСТ» вычислительном кластере.
- Проведена автономная верификация контроллера прерываний. Достигнуто покрытие в 97% по строкам. В процессе верификации было обнаружено 4 ошибки в спецификации устройства и 5 ошибок в эталонной модели
- Система регрессионного прогона используется в работе сектора «Автономная верификация» АО «МЦСТ» как в процессе верификации различных устройств, так и для их регрессионного тестирования

Спасибо за внимание!