

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

Выпускная квалификационная работа бакалавра

# Разработка модулей сопряжения функциональной модели PCI-устройства с RTL-моделью КПИ2

Студент: Неделько Н.В., 913 группа

Научный руководитель: Волконский В.Ю.

# Введение

- на данный момент в составе КПИ2 разрабатываются контроллеры PCI и PCIe, требующие верификации
- для верификации PCI и PCIe контроллеров необходимо моделировать RTL-описание системы с подключенными PCI-устройствами
- повысить скорость и качество верификации можно путем внедрения «гибридных» тестовых систем - содержащих как RTL-модели, так и функциональные модели в тестовом окружении
- в составе эталонных моделей системы архитектуры Эльбрус уже реализованы функциональные модели PCI-устройств (на языке высокого уровня)

Для создания гибридной тестовой системы для КПИ2 необходимо разработать модули сопряжения между RTL-описанием контроллера и функциональными моделями PCI- и PCIe- устройств

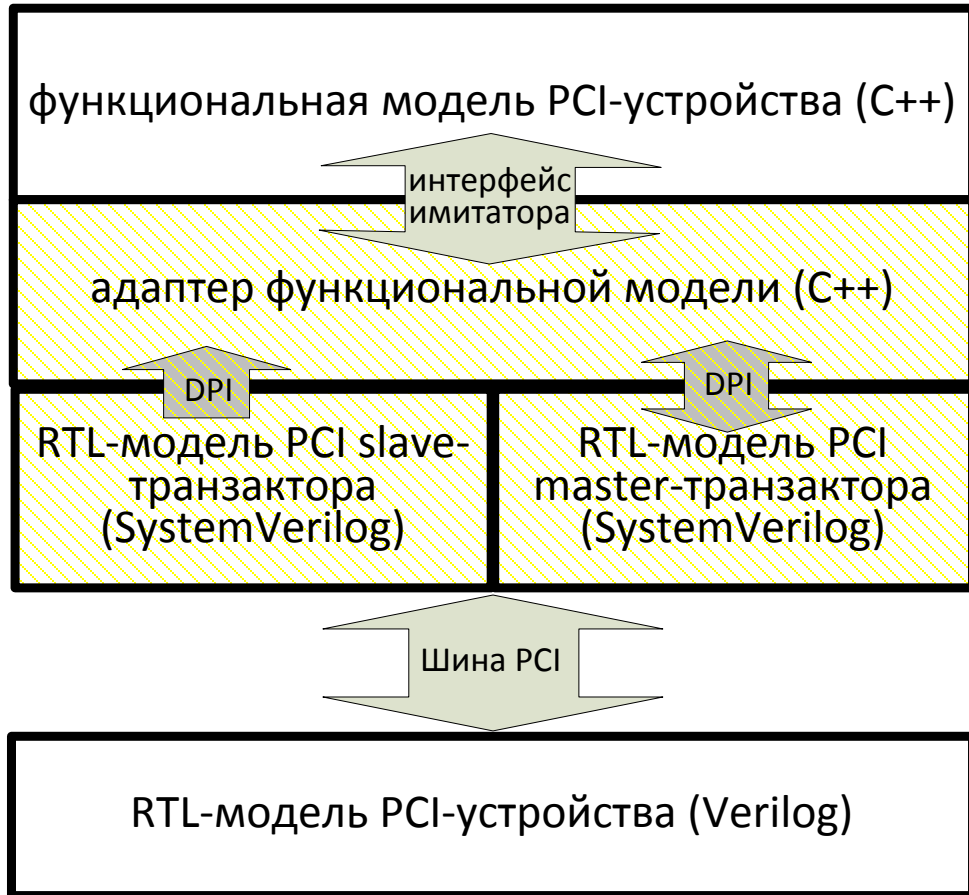
# ПОСТАНОВКА ЗАДАЧИ

Разработать модули сопряжения между функциональной моделью специального тестового PCI-устройства (далее PCI-имитатор) и RTL-описанием КПИ2,

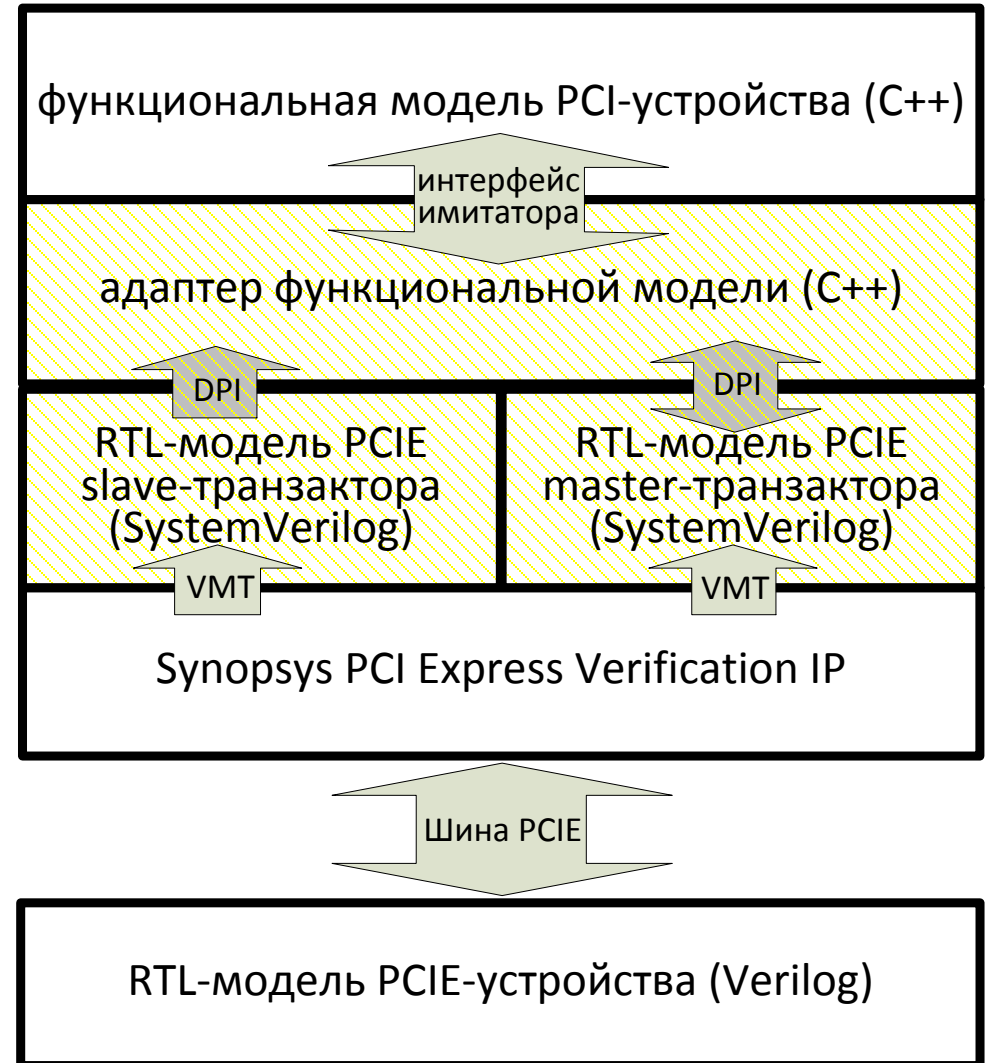
- позволяющие подключать функциональную модель к интерфейсам PCI и PCI express
- поддерживающие всю функциональность PCI-имитатора
- обеспечивающие параметризуемость характеристик:
  - разрядность адресов и данных PCI-транзактора
  - время отклика PCI-slave-устройства
  - поддержка моделью реакций на ошибки четности

# модули сопряжения архитектура

## с PCI-контроллером



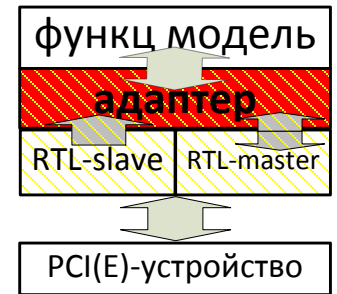
## с PCI Express-контроллером



# МОДУЛИ СОПРЯЖЕНИЯ

## Адаптер функциональной модели

Приспосабливает программный интерфейс функциональной модели к унифицированному программному интерфейсу, построенному на System Verilog DPI.

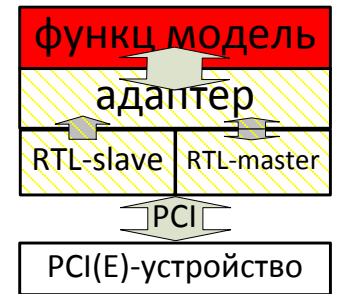


- Создает экземпляры функциональной модели и связывает их со сторонними функциями (которые ими используются, но не реализованы в них).
- Преобразует последовательности событий в RTL-моделях в транзакции функциональной модели
- Транслирует вызовы из унифицированного программного интерфейса в экземпляры соответствующих функциональных моделей.
- Временно хранит данные транзакций.
- Выполняет отладочные функции (трассировка вызовов)

# МОДУЛИ СОПРЯЖЕНИЯ

## Доработка функциональности PCI-имитатора

При разработке модулей сопряжения были выработаны требования по расширению функциональности PCI-имитатора



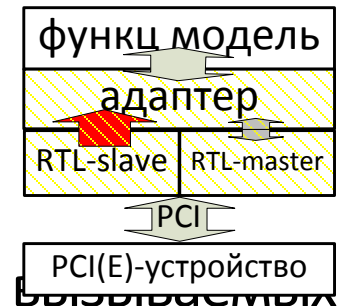
- **Функция для проверки принадлежности данного адреса к какому-либо адресному пространству имитатора.**  
В потактовой модели PCI-транзакция начинается с фазы адреса. Master выставляет адрес на шину и все slave-устройства на ней анализируют его. Ответить должно только то устройство, к адресному пространству которого принадлежит этот адрес.
- **Функция для отправки сообщения имитатору об окончании запрошенной им master-транзакции.**  
Решает 2 проблемы:
  - проблема невозможности одновременной обработки master- и slave- транзакций, если ждать завершения master-транзакции
  - проблема неодновременности(в симуляторе) отправки запроса на master-транзакцию и прихода ответа на нее

# МОДУЛИ СОПРЯЖЕНИЯ

## Унифицированный программный интерфейс: «адаптер функциональной модели – slave-транзактор»

Ведомое устройство PCI или PCIe получает\выдает данные в\из одно из адресных пространств, если последнее содержит указанный при запросе адрес.

Набор функций адаптера функциональной модели, из slave-транзакторов согласно стандарту SystemVerilog DPI.

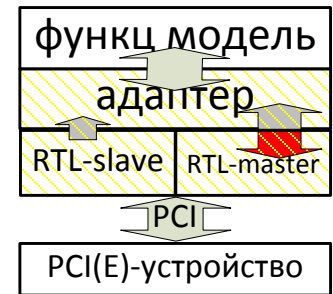


- **slave\_load**: чтение непрерывной последовательности байт из адресного пространства экземпляра функциональной модели  
Параметры: размер и адрес начало области данных, идентификаторы адресного пространства и экземпляра функциональной модели.
- **slave\_store**: запись непрерывной последовательности байт из памяти RTL-модели slave-транзактора в адресное пространство экземпляра функциональной модели.  
Параметры: те же, что и у load.
- **check\_addr\_space**: проверка принадлежности адреса к адресному пространству экземпляра функциональной модели.  
Параметры: адрес для проверки, идентификаторы адресного пространства и экземпляра функциональной модели.

# МОДУЛИ СОПРЯЖЕНИЯ

## Унифицированный программный интерфейс: «адаптер функциональной модели – master-транзактор», DPI-export

Ведущее устройство PCI или PCIe запрашивает чтение или запись и ждет реакции ведомого. Также оно объявляет статус прерывания. Чтобы RTL-часть могла представлять ведущее устройство ей необходимо управлять из адаптера функциональной модели.



В master-транзакторах реализованы функции (SystemVerilog **task**), вызываемые из адаптера согласно спецификации SystemVerilog DPI

- **master\_task\_set** передача master-транзактору задания на создание транзакции от функциональной модели. Задание содержит следующие поля:
  - значимость – равна 1, если задание актуально (еще не завершено)
  - тип – чтение или запись
  - адрес
  - размер передаваемых данных
- **master\_set\_wr\_data** – передача данных для записи в master-транзакции из адаптера функциональной модели в RTL-модель транзактора.
- **master\_set\_int** – передача задания на изменение статуса прерывания INTA#

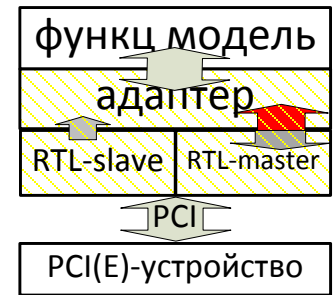


# МОДУЛИ СОПРЯЖЕНИЯ

## Унифицированный программный интерфейс:

«адаптер функциональной модели – master-транзактор», DPI-import

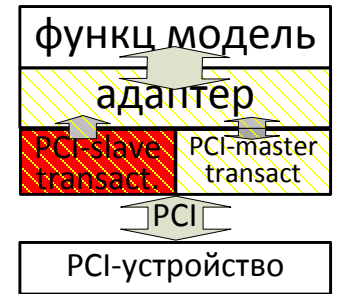
При работе master-транзактор должен сообщить в функциональную модель о завершении транзакции. Для этой и других целей реализован набор функций адаптера, предоставляемых RTL-моделям master-транзакторов согласно спецификации SystemVerilog DPI.



- **master\_begin/master\_end** – инициируют действия адаптера функциональной модели, необходимые на момент начала/конца master-транзакции.
- **update\_read\_buffer** – переписывание данных, принятых в master-транзакции чтения, во временный буфер в адаптере.
- **create\_pci\_imitator** – вызывается в нулевой момент симуляторного времени RTL-моделями каждого master-транзактора, чтобы создать экземпляр PCI-имитатора, передав ему идентификатор и указатели на сторонние функции (из DPI-экспорта).

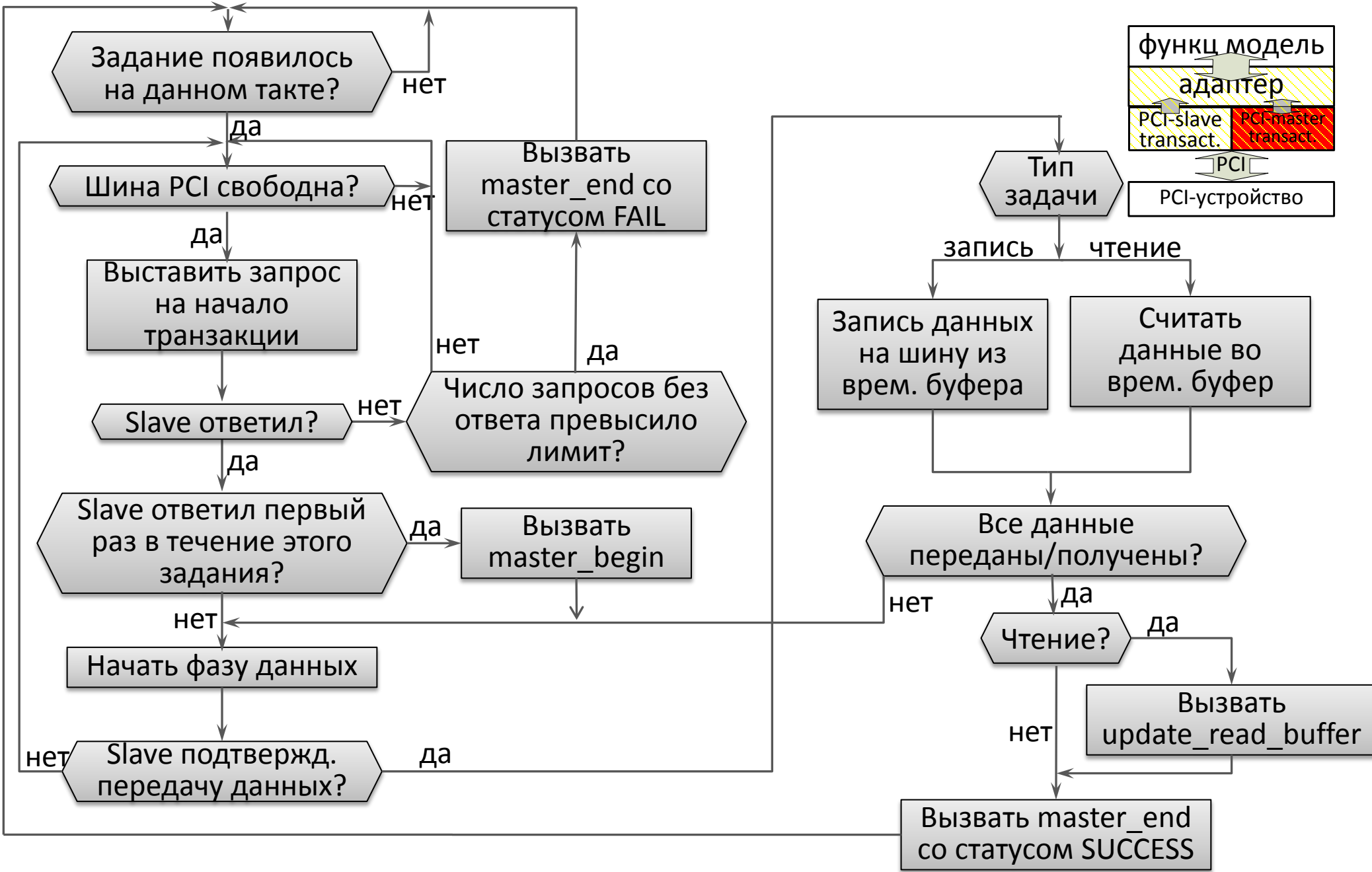
# МОДУЛИ СОПРЯЖЕНИЯ

## Алгоритмы работы транзакторов PCI-slave



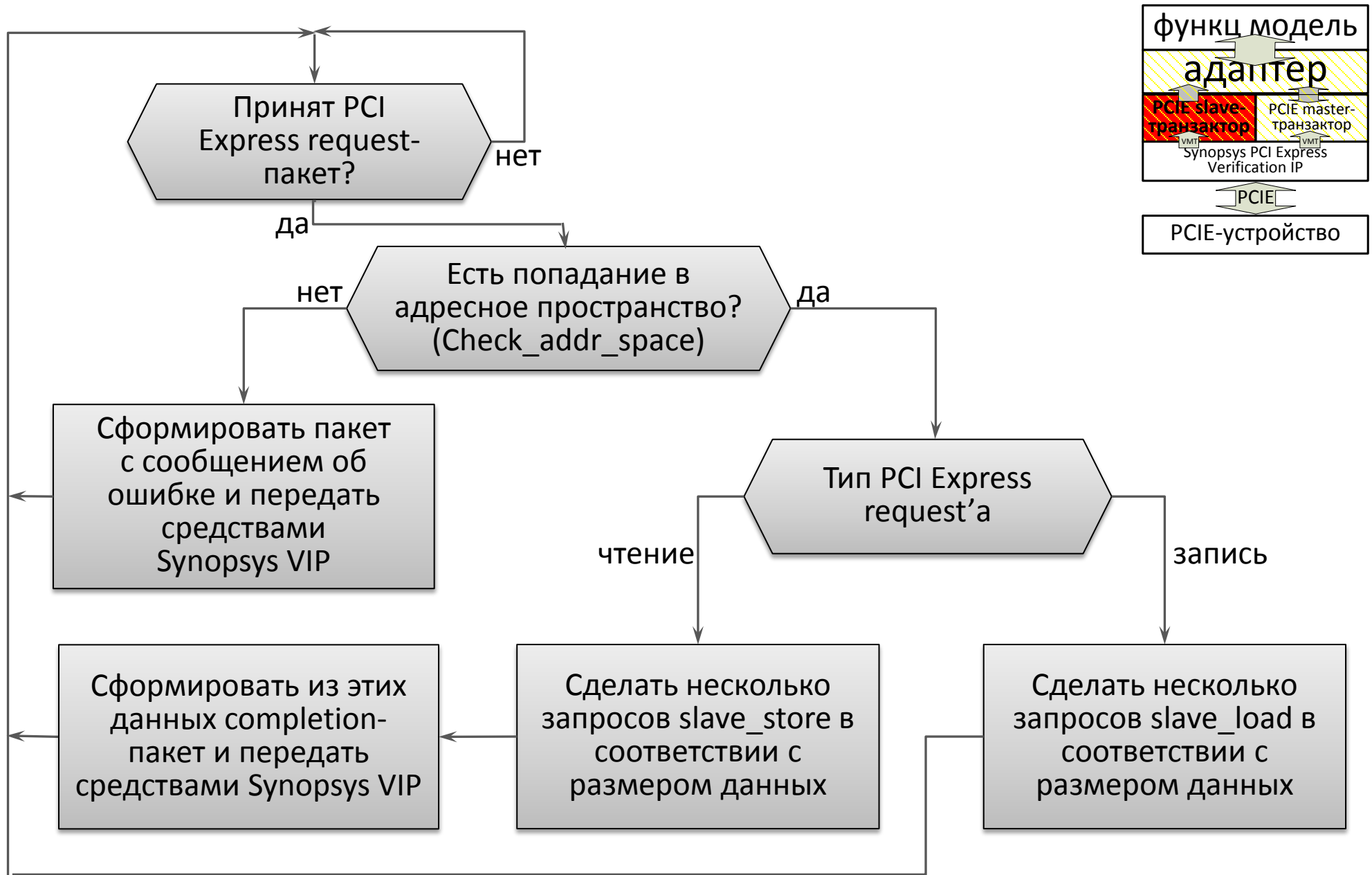
# МОДУЛИ СОПРЯЖЕНИЯ

## Алгоритмы работы транзакторов PCI-master



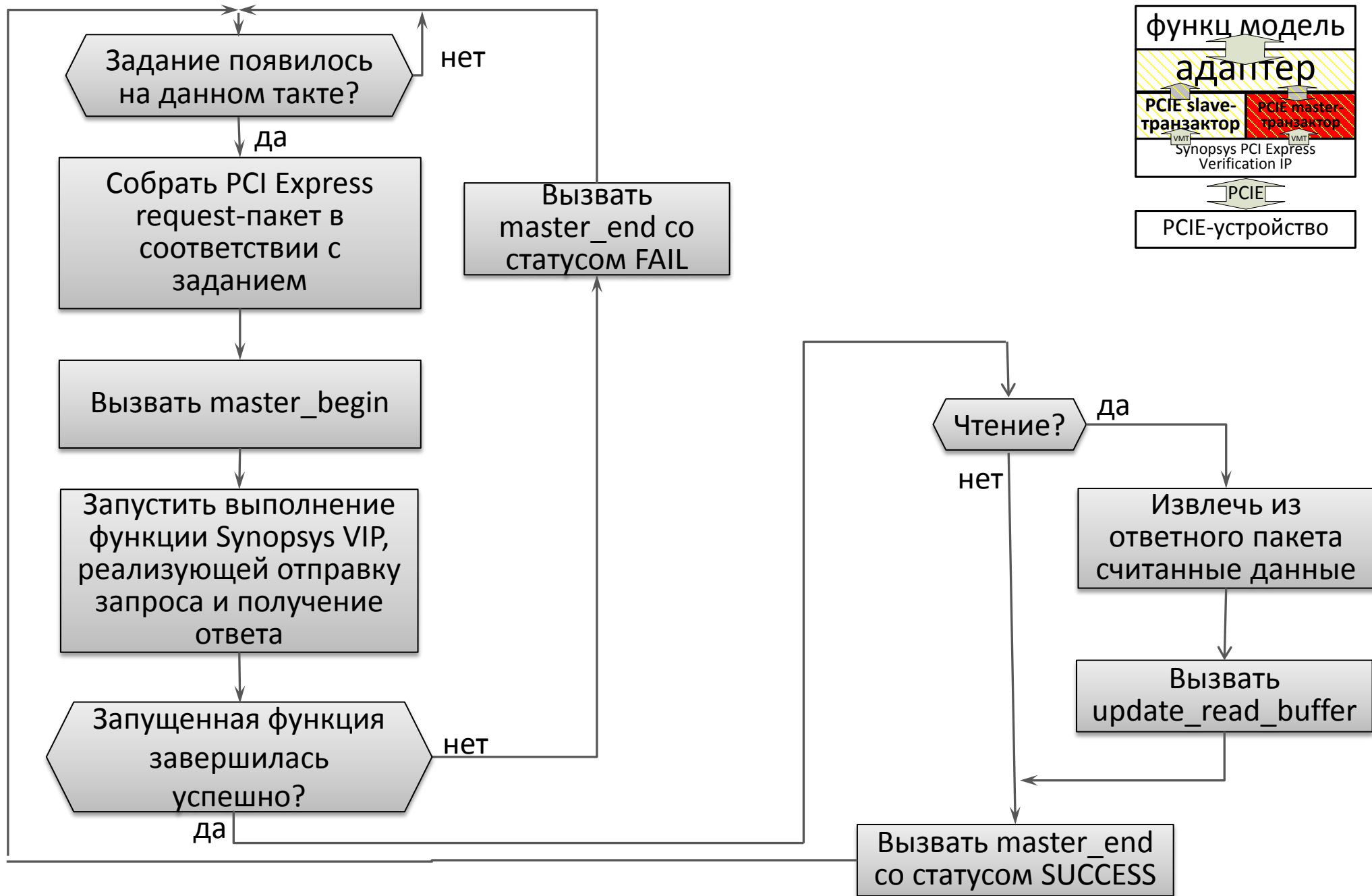
# МОДУЛИ СОПРЯЖЕНИЯ

## Алгоритмы работы транзакторов PCI Express slave



# МОДУЛИ СОПРЯЖЕНИЯ

## Алгоритмы работы транзакторов PCI Express master



# МОДУЛИ СОПРЯЖЕНИЯ

## Дополнительная функциональность RTL-моделей транзакторов

Позволяет моделировать PCI-клиенты, различающиеся по характеристикам, вариацию которых допускает стандарт PCI с целью повышения качества верификации (проверка различных режимов взаимодействия, реакции на клиенты с различными временами отклика и т.д.).

- Возможность включения поддержки 64-разрядных адресов в PCI-транзакторах
- Возможность включения поддержки 64-разрядных данных в PCI-транзакторах
- Возможность моделирования времени отклика PCI-slave-устройства
- Возможность включения поддержки каждым из PCI-транзакторов реакций на ошибки четности

# Результаты работы

- Разработаны модули сопряжения между RTL-моделями PCI\PCIE контроллеров в составе КПИ2 и функциональной моделью PCI-имитатора (на языках C++ и System Verilog).
- Реализована следующая дополнительная функциональность:
  - выбор разрядности адресов и данных PCI-транзактора, как параметра
  - выбор времени отклика PCI-slave-устройства, как параметра
  - выбор реакции PCI-транзакторов на ошибки четности, как параметра
- Найдены ошибки в 5 инженерных тестах, направленных на верификацию КПИ2
- Выработаны рекомендации по модификации\разработке функциональных моделей устройств для работы в составе гибридных тестовых систем.

# Результаты работы

## Возможности для доработки и использования результатов в других задачах

- Сопряжение других функциональных моделей имеющих программный интерфейс как у PCI-имитатора
- Доработка адаптера функциональных моделей для поддержки функциональной модели с программным интерфейсом, отличным от интерфейса PCI-имитатора
- Создание других транзакторов с разработанным мной унифицированным DPI-интерфейсом
- Введение переменных (возможно, случайных) задержек ожидания готовности в PCI-транзакции (это даже предусмотрено в коде транзакторов)
- Вариирование большего числа характеристик транзакторов