

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

Выпускная квалификационная работа
(магистерская диссертация)

Методология использования компонентов функциональной модели ВК для верификации контроллеров южного моста

Студент: Неделько Н.В., 913 группа

Научный руководитель: к. т. н. Мешков А.Н.

Введение

При верификации контроллеров южного моста требуется создание двух моделей периферийного устройства (ПУ):

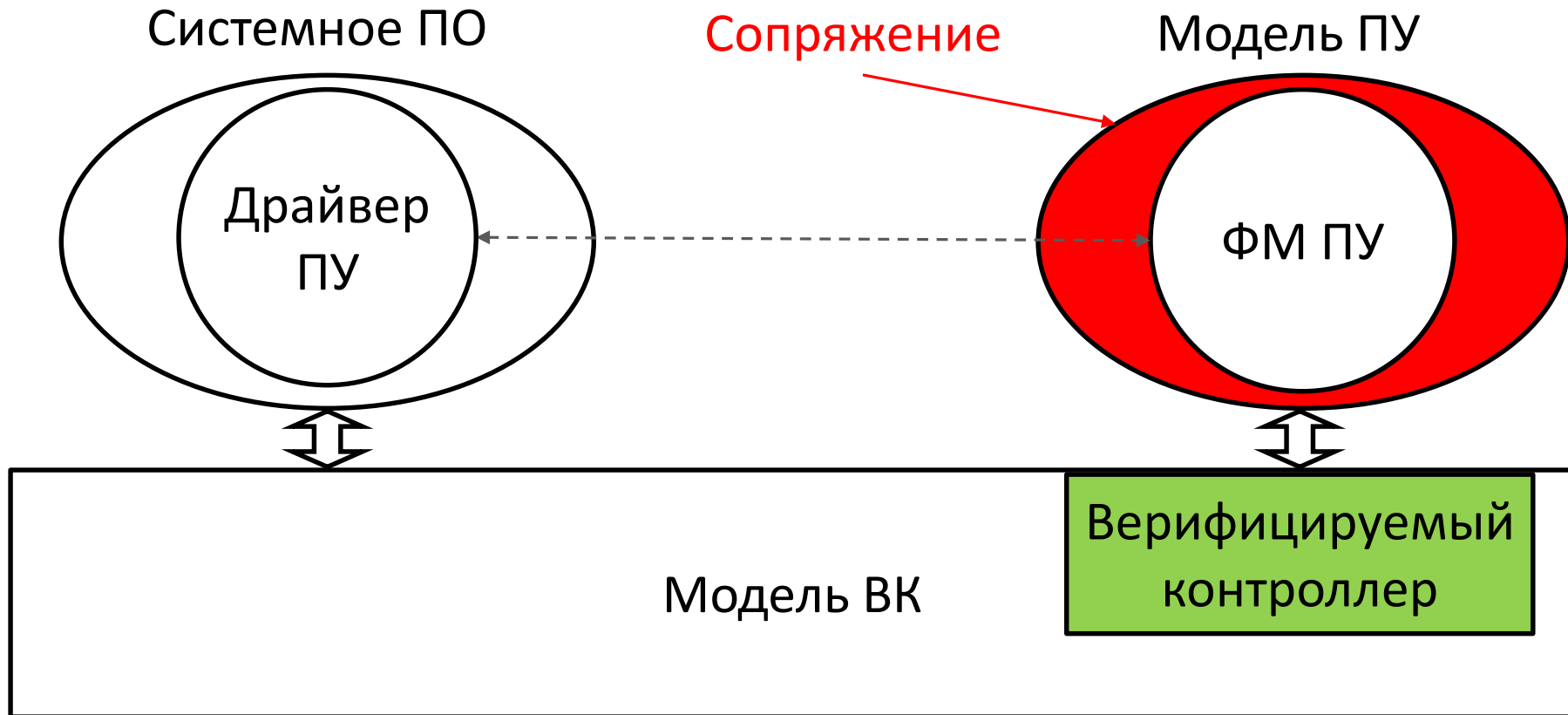
- для подключения к RTL-модели верифицируемого контроллера (DUT) – в *беспроцессорной сборке*
- для подключения к функциональной модели верифицируемого контроллера (ФМ DUT) – в *Симуляторе*.

Работа над созданием таких моделей может быть сокращена за счет сопряжения функциональной модели(ФМ) периферийного устройства с RTL-моделью верифицируемого контроллера вместо написания RTL-модели периферийного устройства или покупки VIP.

Симулятор – ФМ ВК с подключенными периферийными устройствами
Беспроцессорная сборка – тестовое окружение для RTL-модели южного моста, управляющееся тестами на C++, подключенное к периферийным устройствам и к имитатору процессора.

Введение

Схема сопряжения



- Один драйвер может управлять несколькими экземплярами функциональной модели
- Возможно моделирование одновременной работы нескольких различных драйверов

Постановка задачи

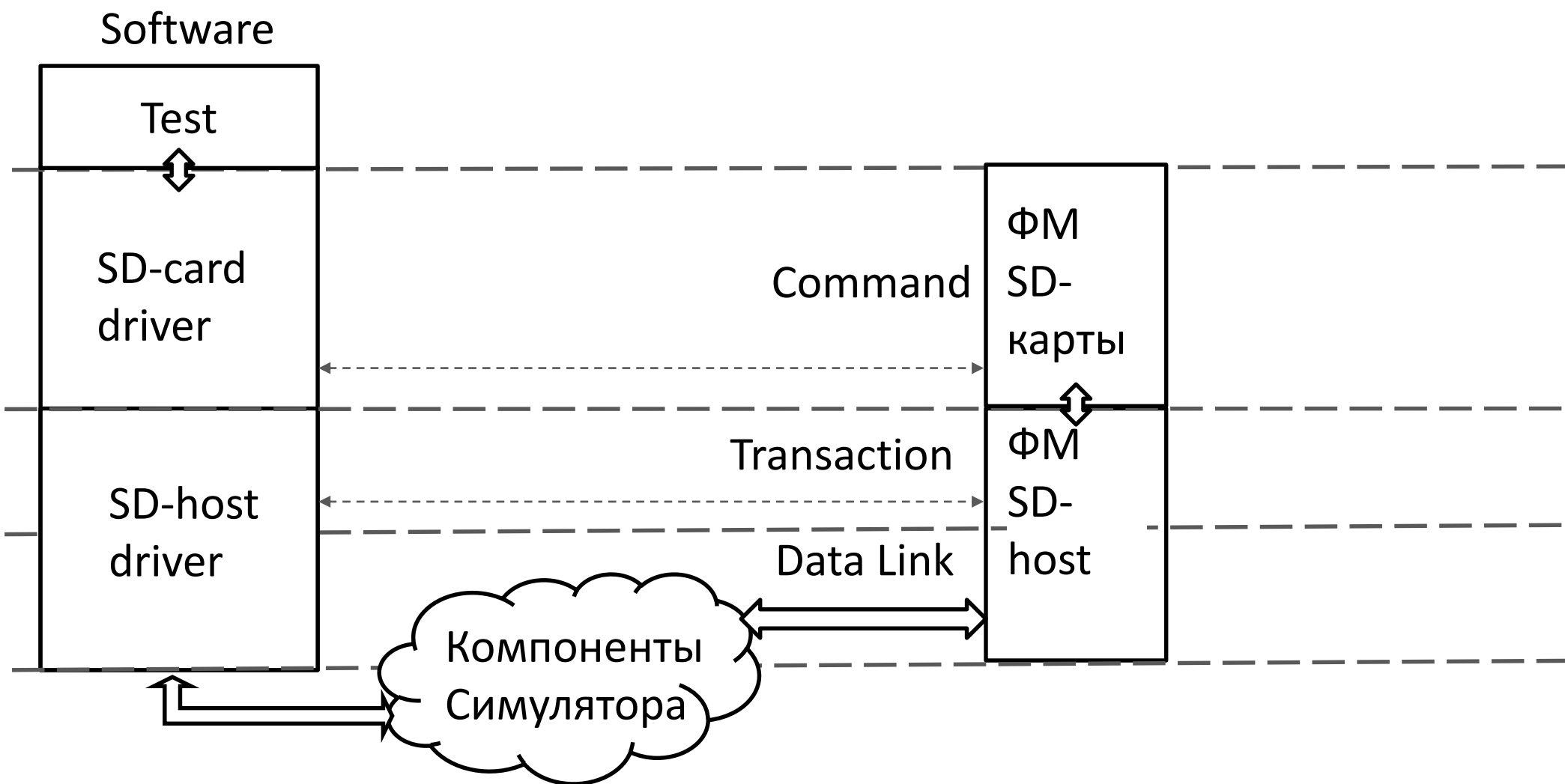
- разработать модули сопряжения между функциональными моделями устройств и соответствующими им контролерами южного моста:
 - PCI(master/target) <=> PCI(target/master)
 - PCI(master/target) <=> PCIE(slave/master)
 - USB mass storage device <=> USB 2.0 host
 - SD memory card <=> SD-host
 - ATAPI SCSI Media device <=> SATA host
- Разработать тестовый драйвер устройства «SD-memory card»

Уровни стека протоколов взаимодействия компонентов АПК с периферийными устройствами

	SATA	PCIE	PCI	SD	USB
Physical Layer(RTL)	«Phy layer»	«Physical layer»	RTL-модель тривиальна ввиду синхронности интерфейса		«Electrical: Signaling»
Data Link layer	формирование и пересылка фреймов с контролем целостности				
	«Link layer»	«Data Link Layer»	«...:Basic Transfer Control» Фреймы: 32/64-битные	«SD Memory Card System Concept: bus protocol». Фреймы: Блоки определенной длины	«Protocol layer: 1-4» Фреймы: блоки определенной ограниченной длины
Transaction layer	Различный обмен данными: представление данных в виде пакетов, реакция на ошибки				
	«Transport layer», «device command layer»	«Transaction Layer»	Target:Имитация доступа в память (11 транзакций), Master: прием/передача последов-ти байт	Обмен команда-ответ, прием /отправка последовательностей байт данных (2 типа)	«Protocol layer»: 5-7, USB Data Flow Model bulk/ interrupt/ control/ isochronous Transfer (4 типа)
Command layer	ATA commands, SCSI commands	Реакция устройства на доступ в определенные байты memory-, configuration- и IO-пространств.		Действия Device в зависимости от пришедшей SD-команды	«USB Device Framework», USB Mass storage Device: Command-data-status protocol

Взаимодействие прикладного ПО с ПУ на примере SD-card

Функциональная модель

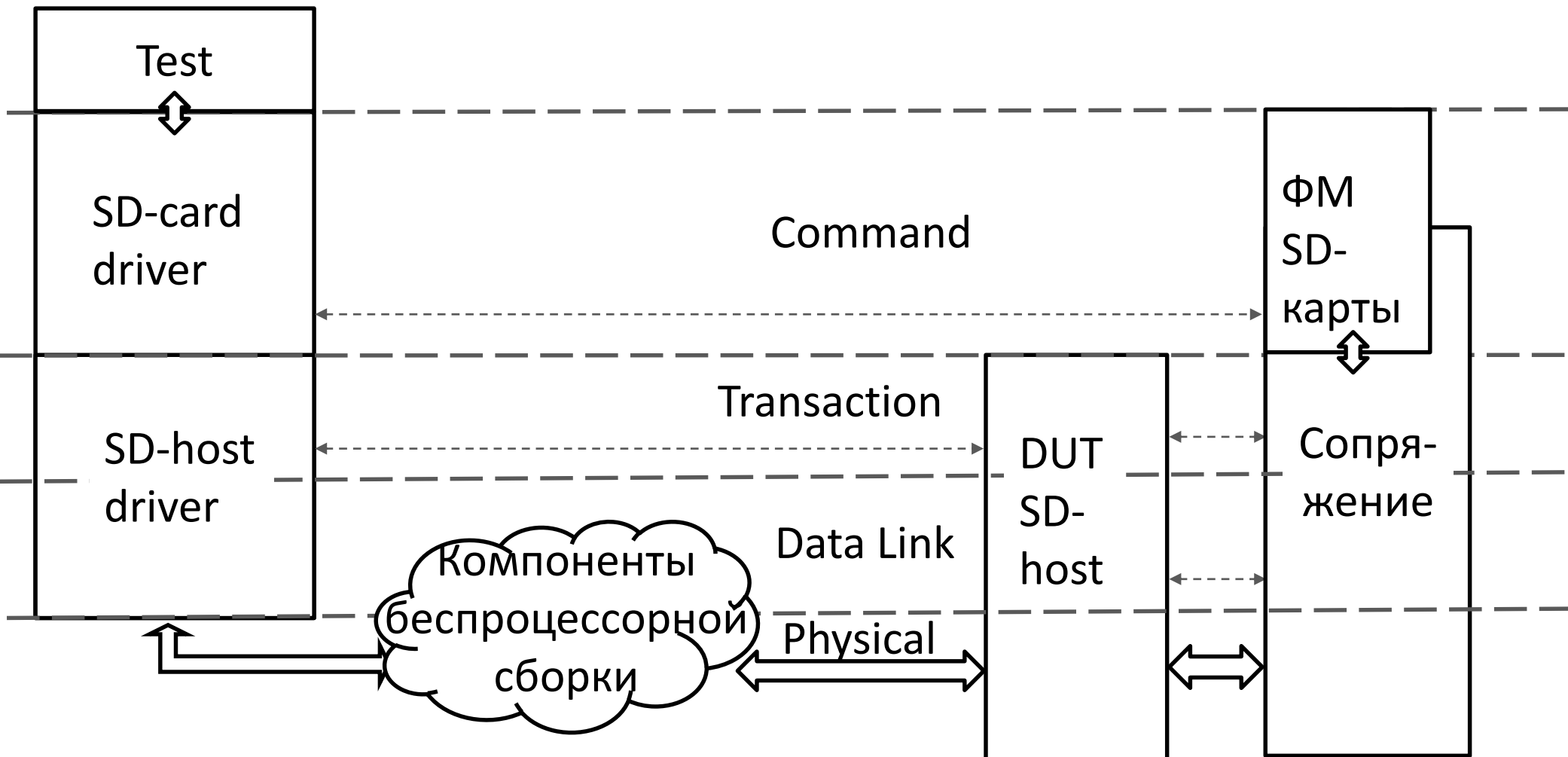


Взаимодействие прикладного ПО с ПУ на примере SD-card

Модель на основе беспроцессорной сборки

Software

Модель SD-карты



Сложности подхода

Получение данных с вышестоящих уровней

Проблема – сопряжению нужна информация, не достижимая через интерфейс ФМ ПУ, но упакованная на уровнях, реализованных в этой ФМ

Причины недостижимости этой информации:

- Она не нужна ФМ ПУ поскольку в модель ее не использует
- Она не нужна ФМ DUT (при работе в составе Симулятора)

Решения:

- Изменение интерфейса ФМ ПУ
- Частичная реализация вышестоящего уровня в сопряжении
 - + Не нужно править ФМ, которая уже давно хорошо отлажена
 - + Можно сделать больше проверок правильности работы ФМ ПУ
 - Возможна повторная реализация функциональности ФМ ПУ

Сложности подхода

Связь сопряжения с тестовым драйвером при использовании беспроцессорной сборки

Проблема – необходимо обеспечить взаимодействие сопряжения с тестовым драйвером, выходящее за рамки программного интерфейса ПУ

- Покрытие DUT требует варьирования программно недоступных свойств ПУ
- Для ускорения моделирования требуется запуск упрощенного прохода определенных стадий работы устройства
- Требуется независимая от DUT проверка корректности передачи данных через DUT.

Решения:

- Прямой вызов функций сопряжения или ФМ ПУ из тестового драйвера
- Использование зарезервированных команд для передачи информации сопряжению через DUT

Типичная структура сопряжения

К функциональной модели



C++

Command Layer

DPI

Transaction Layer

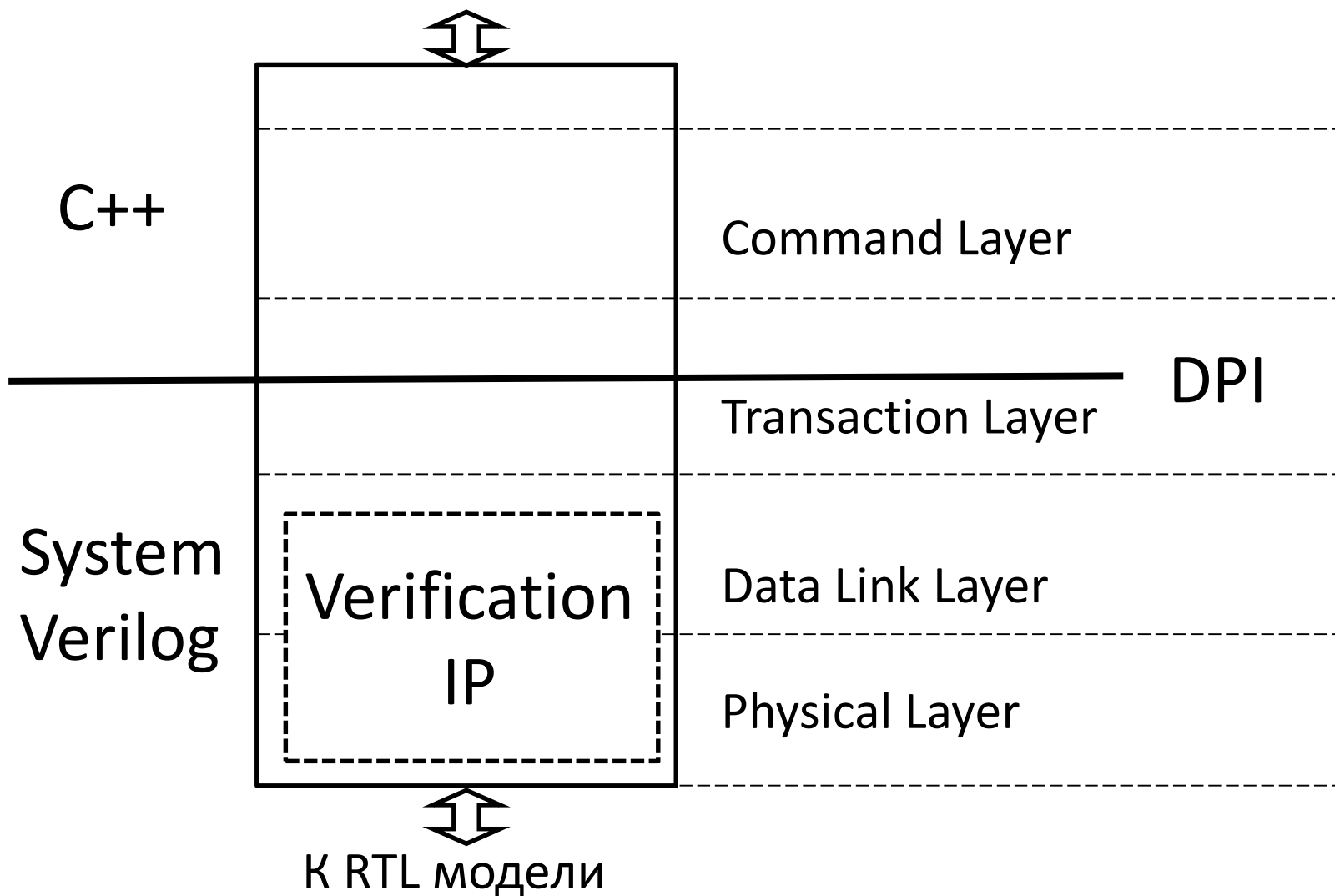
System
Verilog

Verification
IP

Data Link Layer

Physical Layer

К RTL модели



Средства верификации до начала работы

Компонент	PCI master\target	PCIE master\slave	USB host	SD host	SATA host
Сопряжение: Physical & Data Link Layers	Нет	VIP	VIP	Нет	VIP
Сопряжение: Transaction Layer	Нет	Нет	Нет	Нет	Нет
Сопряжение: Command Layer	ФМ	ФМ	ФМ	ФМ	ФМ
ФМ DUT	Разработана	Разработана	Разработана	Нет	Разработана
Тестовый драйвер ПУ	Разработан	Разработан	Разработан	Нет	Разработан
ФМ ПУ	Разработана	Разработана	Нет	нет	Разработана

Особенности реализации сопряжений DUT: PCI/PCIE(slave/master)

Physical Layer		Data Link Layer		Transaction Layer	Command Layer
PCI	PCIE	PCI	PCIE		
Verilog, синхронный параллельный	VIP, асинхронный Последовательный	Verilog	VIP	C++, В PCI – только master	В ФМ

Общие сложности:

- Сопряжение не декодирует уровень команд: многоцелевая функция в интерфейсе – требуя аргумент для одной цели, выдает результат для другой цели, а он нужен более низкому уровню **до возможности вычислить аргумент.**

Решение – выделить первую цель в функцию `check_tet_sрасе`

Специфичные сложности:

- Необходимость отслеживания корректности работы интерфейса PCI.

Решение – PCI-monitor, исчерпывающе анализирующий сигналы управления на шине.

- Неподготовленность ФМ ПУ к вызовам **verilog-tasks** с **ненулевым временем симуляции**

Решение – модификация ФМ ПУ

Особенности реализации сопряжений DUT:USB 2.0 Host

Physical Layer	Data Link Layer	Transaction Layer	Command Layer
VIP асинхронный последовательный	VIP	SV/C++	C++, частично

Общие сложности:

- Сопряжение не декодирует уровень команд => ему не доступна необходимая информация для вычисления размера запрашиваемых из модели данных
Решение – частичная реализация в сопряжении уровня команд.
- Требуется связь сопряжения с тестовым драйвером: необходимо имитировать присоединение USB-устройства к разъему
Решение – прямой вызов из драйвера соответствующей функции сопряжения

Особенности реализации сопряжений DUT: SD-host

Physical Layer	Data Link Layer	Transaction Layer	Command Layer
SV, синхронный параллельный (шина данных параллельно с линией команд)	SV	C++/SV	C++, частично

Общие сложности

- Сопряжение не декодирует уровень команд => ему не доступны необходимые данные о
 - ширине шины данных
 - длине блока данных для data-link-уровня
Решение – модификация ФМ ПУ
 - признаке последнего пакета
Решение – частичная реализация в сопряжении уровня команд
- Необходимость нестандартного взаимодействия с тестовым драйвером
 - Запуск инициализации карты, идущей без участия хоста
Решение – через зарезервированную команду
 - Сверка возвращаемых картой значений с полученными от хоста
Решение – функция сопряжения, доступная тестовому драйверу

Особенности реализации сопряжений

DUT: SATA-host

Physical Layer	Data Link Layer	Transaction Layer	Command Layer
VIP асинхронный последовательный	VIP	SV/C++	C++, частично

Общие сложности

- Сопряжение не декодирует уровень команд => ему не доступны необходимые данные о
 - типе пришедшей SCSI- или ATA- команды
 - «features», установленных в процессе

Решение – частичная реализация уровня команд.

Специфичные сложности

- Необходимость установления соответствия между экземплярами модулей в Verilog и C++-объектов ФМ

Решение - Во все экспортируемые Verilog-кодом функции добавлен аргумент “ID”, являющийся для каждого Verilog-модуля параметром, а для C++-объекта – индексом в массиве.

Средства верификации сейчас

Компонент	PCI master\target	PCIE master\slave	USB host	SD host	SATA host
Physical & Data Link Layers		VIP	VIP		VIP
Transaction Layer					
Command Layer	В ФМ	В ФМ	частично	частично	частично
ФМ DUT	Разработана ранее	Разработана ранее	Разработана ранее		Разработана ранее
Тестовый драйвер ПУ	Разработан ранее	Разработан ранее	Разработан ранее		Разработан ранее
ФМ ПУ в составе симулятора	Разработана ранее	Разработана ранее			Разработана ранее

 Реализовано мной

 Реализовано в секторе моделирования за 2 года

* Все ФМ ПУ и тестовые драйверы, разработанные ранее, модифицировались в процессе разработки сопряжений

Результаты работы

- Разработаны модули сопряжения (на языках C++ и System Verilog) между функциональными моделями периферийных устройств и RTL-моделями соответствующих им контроллеров южного моста.
- Реализован тестовый драйвер SD-карты.
- Созданы тесты SD-host-контроллера, в результате чего найдены ошибки:
 - SD-host RTL: 47
 - SD-host DOC: 4
 - SD-host ФМ: 21
 - SD-card ФМ: 10