

Приложение Platform Designer

Цели

Цели – научиться:

- Создавать системы используя Platform Designer (PD)
- Интегрировать существующие IP в систему используя PD
- Использовать систему, созданную в PD в пакете Intel® Quartus® Prime (QP)
- Разрабатывать и использовать компоненты (custom IP) при создании системы в PD
- Моделировать систему, созданную в PD, с помощью ModelSim (используя NativeLink QP)
- Отлаживать систему, созданную в PD, на плате с помощью InSystem Source&Probe Editor и SignalTapII пакета QP

Три варианта пакета Intel® Quartus® Prime

Intel® Quartus® Prime

Design Software

НЕ требуется лицензия

Lite Edition (LE)

Требуется лицензия

Standard Edition (SE)

Требуется лицензия

Pro Edition (PE)

Для выполнения лабораторных работ достаточно:

- QPLite
 - ModelSim Intel FPGA **Starter** Edition
- Версия пакета QP – любая, начиная с 16.0

[Сравнение версий на сайте Intel® FPGA](#)

Приложение Platform Designer

Часть 1

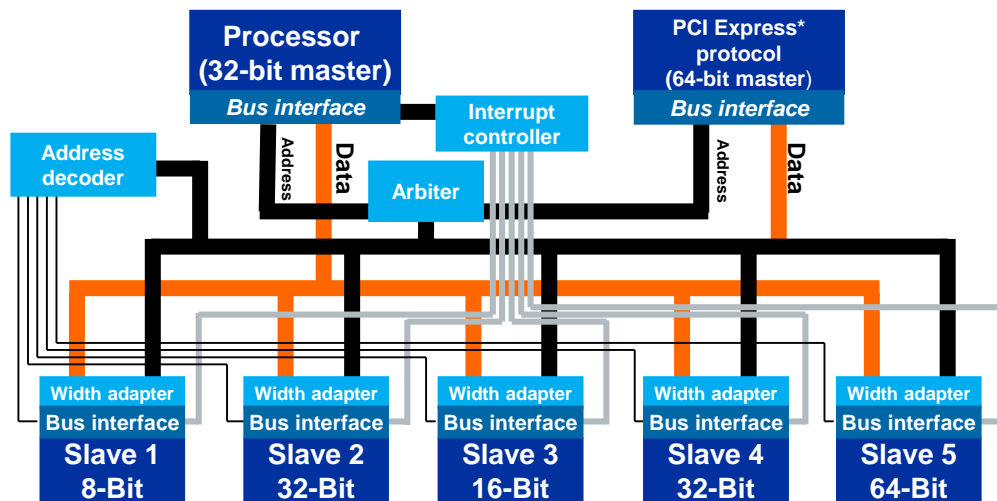
Пользовательский интерфейс

План

- О приложении PD
- Пользовательский интерфейс приложения PD
- Файлы, создаваемые PD, при генерации системы
- Место PD в процедуре проектирования пакета QP
- Лабораторная 1

Традиционная процедура разработки системы

- Компоненты могут использовать разные интерфейсы (некоторые – стандартные, некоторые – нестандартные)
- Значительные усилия требуются для разработки системы коммуникации между компонентами системы.
- Интеграция блоков в систему требует длительной отладки



PD: Автоматическое создание системы соединения

- Сокращает время разработки: автоматическое создание необходимых блоков и связей
- Позволяет избежать ошибок
- Позволяет сфокусироваться на разработке архитектуры системы и собственных блоков
- Повышает производительность труда разработчика.



Преимущества использования PD

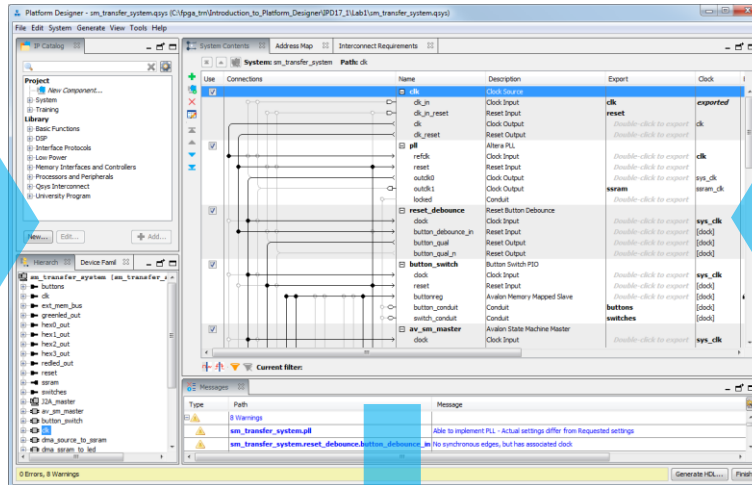
- Упрощает создание сложных систем: автоматизирует процесс создания системы межсоединений
- Позволяет работать на более высоком уровне абстракции: на уровне системы и связей в системе, а не на уровне сигналов
- Инструмент для интеграции в систему: стандартных IP, пользовательских IP, процессорных элементов, средств моделирования
- Позволяет использовать созданные системы как компоненты (иерархическое проектирование на системном уровне)
- Позволяет сократить время разработки и упростить верификацию

Простота использования PD



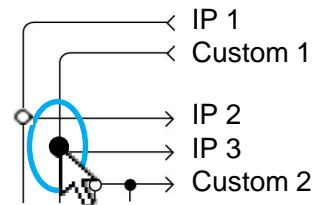
- Interface protocols
- Memory
- DSP
- Embedded
- Bridges
- PLL
- **Custom systems**

Библиотеки компонентов



HDL

Connect custom IP and systems



*Графический
интерфейс для
создания системы*

Автоматические: интеграция и создание HDL описания

Конечные приложения для использования PD

PD может использоваться при создании любого проекта для FPGA

- Два типа взаимодействия в цифровой системе
 - Управление системой (**Control plane**)
 - Адресный доступ (**Memory-mapped**)
 - Чтение и запись регистров управления и статуса
 - Передача данных (**Data plane**)
 - Поточковая передача (**streaming data transfer**): высокоскоростная, точка-точка
 - Адресная передача (**Memory-mapped**)
- Приложения:
 - ЦОС
 - Обработка видео потоков
 - Высокоскоростные интерфейсы
 -

Инициаторы взаимодействия в системах PD

Для инициации взаимодействия система, создаваемая в PD, не требует использования процессора

- Для инициирования передачи данных и управления системой могут использоваться существующие IP, пользовательские компоненты, внешние процессоры (контроллеры).
 - Управление системой:
 - компоненты с адресным доступом (Memory-mapped) используют master → slave взаимодействие
 - Примеры: конечный автомат, модули прямого доступа к памяти - direct memory access (DMA)
 - Передача данных:
 - компоненты с адресным доступом (Memory-mapped) используют master → slave взаимодействие
 - компоненты с потоковой передачей (streaming data) используют uses source → sink взаимодействие
 - Примеры: поток данных от видео камеры, АЦП...

Использование процессоров в системах PD

Система, содержащая встроенный ARM (Hard Processor System (HPS)) или soft-core процессор (Intel Nios II), требует использования PD

- Встроенные Hard-core процессор Arm (Cortex-A9) или soft-core процессор NiosII могут подключаться к системе, создаваемой в PD для:
 - Управления
 - используют master → slave взаимодействие
 - Передачи данных
 - используют master → slave взаимодействие

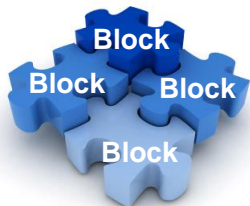
PD - средство проектирования систем

Уровень абстракции описания и производительность при создании системы

Low

Medium

High

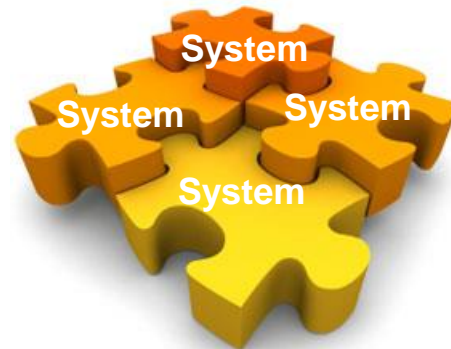


Интеграция блоков



IP интеграция

- Разработка на основе IPs
- повторное использование IP
- верификация IP



Создание системы

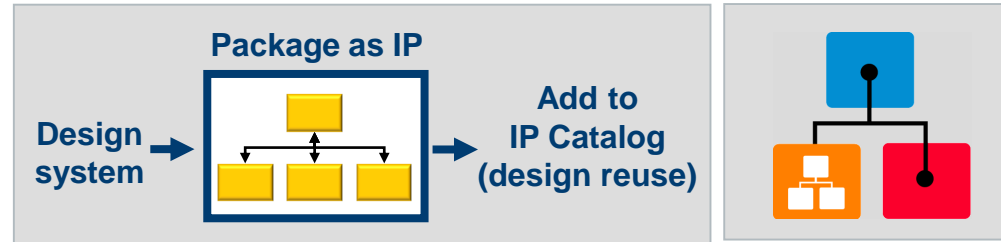
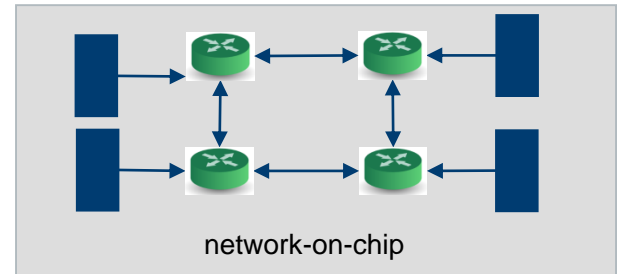
- Иерархические системы
- повторное использование
- верификация

Схемные редакторы и HDL описания

Platform Designer

Особенности PD

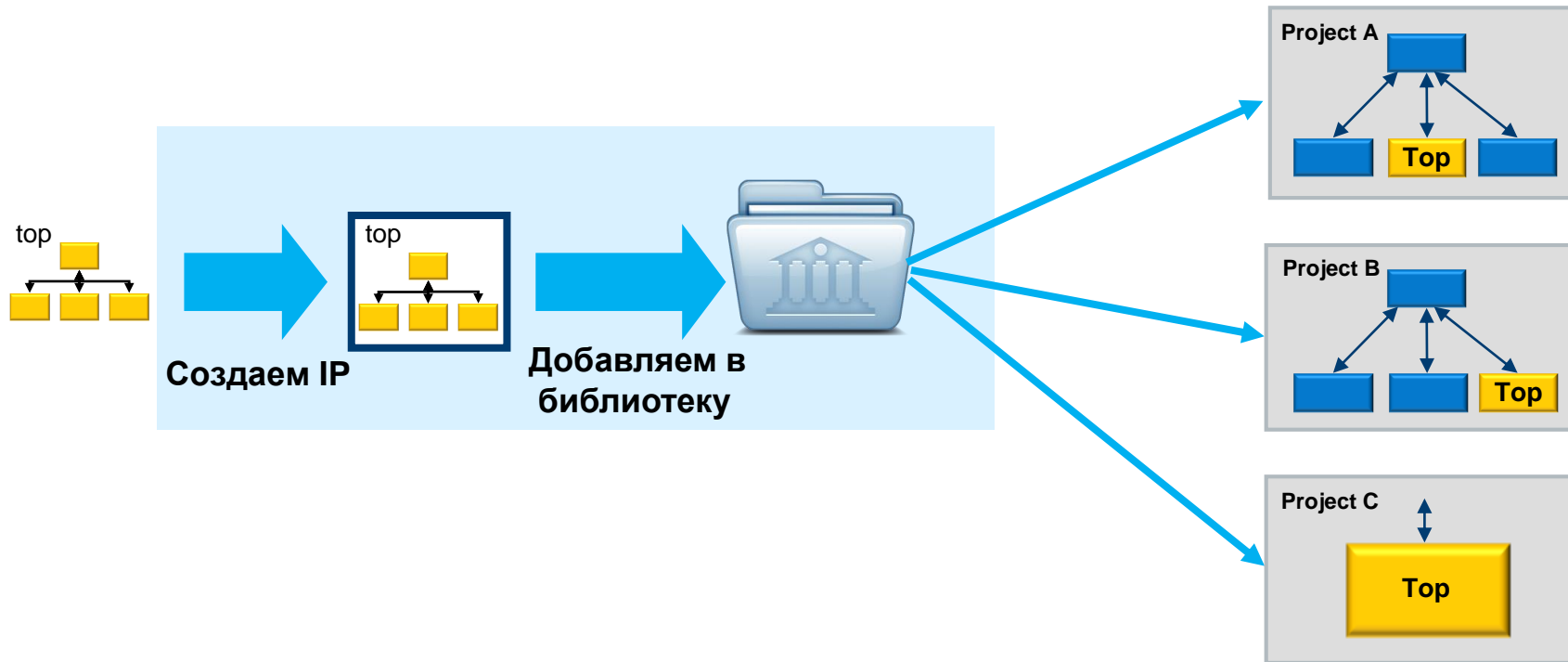
- Высокоскоростные каналы соединения компонентов системы
- «Управление» IP
- Иерархическое проектирование систем
- Поддержка стандартных интерфейсов



Intel® FPGA	Avalon® interface
Arm*	Arm AMBA* AXI interface

Использование созданных систем и компонентов

PD позволяет повторно использовать (re-use) созданные компоненты и системы

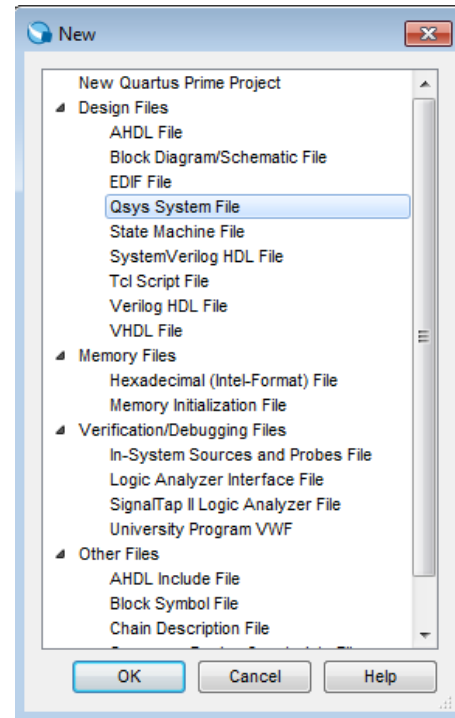
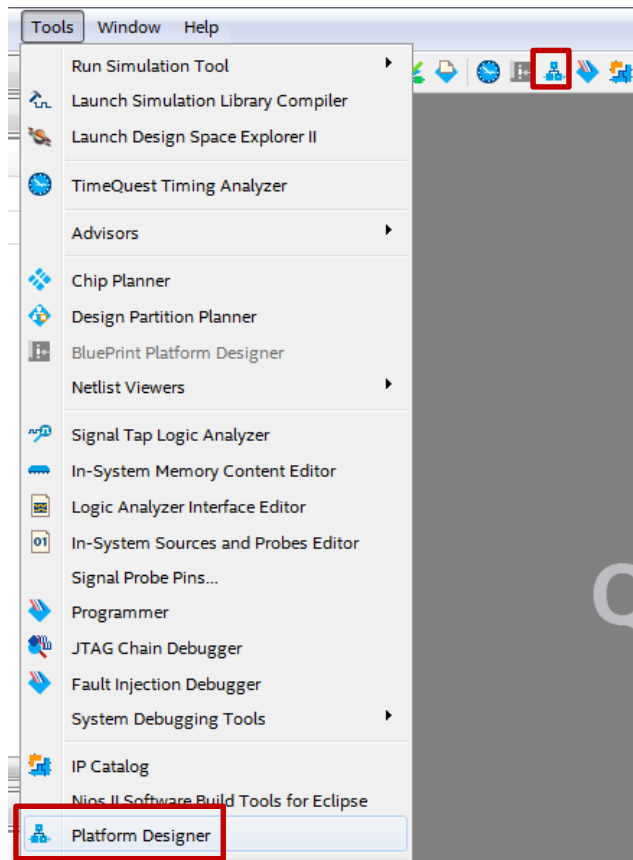
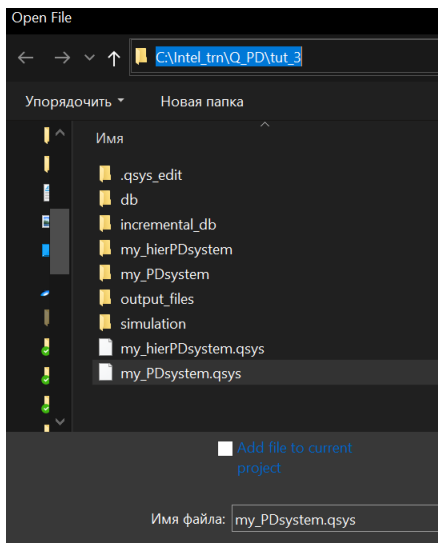


План

- О приложении PD
- Пользовательский интерфейс приложения PD
- Файлы, создаваемые PD, при генерации системы
- Место PD в процедуре проектирования пакета QP
- Лабораторная 1

Как запустить PD

1. QP меню Tools
2. Создать новый файл Platform Designer (.qsys)
3. Открыть существующий файл (.qsys) из пакета Quartus



Пользовательский интерфейс PD

Отсоединяемые закладки
(Detachable Tabs)

Platform Designer - my_PDsystem.qsys* (C:\Intel_trn\Q_PD\tut_3\my_PDsystem.qsys)

File Edit System Generate View Tools Help

IP Catalog

Project

- New Component...
- System
- Training
- Library
 - Basic Functions
 - DSP
 - Interface Protocols
 - Low Power
 - Memory Interfaces and Controllers
 - Processors and Peripherals
 - Qsys Interconnect
 - Tri-State Components
 - University Program

Библиотека IP

New... Edit... Add...

Hierarchy

Device Family

- my_PDsystem [my_PDsystem.qsys*]
 - clk
 - dout_a
 - dout_b
 - reset
 - clk_0
 - my_master_0
 - clock
 - conduit_end_0
 - reset
 - s0
 - my_slave_0
 - conduit_end_0

Иерархия системы

System Contents

Address Map

Interconnect Requirements

System: my_PDsystem Path: my_slaveWS_0.clock

Use	Connec...	Name	Description	Export	Clock	Base	End
<input checked="" type="checkbox"/>		clk_0	Clock Source	clk	exported		
<input checked="" type="checkbox"/>		clk_in	Clock Input	Double-click to	clk_0		
<input checked="" type="checkbox"/>		clk_in_reset	Reset Input	Double-click to	[clock]		
<input checked="" type="checkbox"/>		clk	Clock Output	Double-click to	[clock]		
<input checked="" type="checkbox"/>		clk_reset	Reset Output	Double-click to	[clock]		
<input checked="" type="checkbox"/>		my_master_0	my_master	Double-click to	clk_0		
<input checked="" type="checkbox"/>		clock	Clock Input	Double-click to	[clock]		
<input checked="" type="checkbox"/>		reset	Reset Input	Double-click to	[clock]		
<input checked="" type="checkbox"/>		m0	Avalon Memory Mapped ...	Double-click to	clk_0		
<input checked="" type="checkbox"/>		my_slave_0	my_slave	Double-click to	clk_0		
<input checked="" type="checkbox"/>		clock	Clock Input	Double-click to	[clock]		
<input checked="" type="checkbox"/>		reset	Reset Input	Double-click to	[clock]		
<input checked="" type="checkbox"/>		s0	Avalon Memory Mapped ...	Double-click to	clk_0		
<input checked="" type="checkbox"/>		conduit_end_0	Conduit	Double-click to	[clock]		
<input checked="" type="checkbox"/>		my_slaveWS_0	my_slaveWS	Double-click to	unconnected		
<input checked="" type="checkbox"/>		clock	Clock Input	Double-click to	[clock]		
<input checked="" type="checkbox"/>		reset	Reset Input	Double-click to	[clock]		
<input checked="" type="checkbox"/>		s0	Avalon Memory Mapped ...	Double-click to	[clock]		
<input checked="" type="checkbox"/>		conduit_end_0	Conduit	Double-click to	[clock]		

System Contents (структура системы)

Current filter:

Parameters

System: my_PDsystem Path: my_slaveWS_0.clock

Clock Input

clock_sink

Details

Parameters

Clock rate: 0

Параметры компонента

Messages

Type	Path	Message
Error	1 Error	
Error	my_PDsystem.my_slaveWS_0	my_slaveWS_0.clock must be connected to a clock output
Warning	1 Warning	
Warning	my_PDsystem.my_slaveWS_0	my_slaveWS_0.s0 must be connected to an Avalon-MM master

Сообщения (info, warning, error)

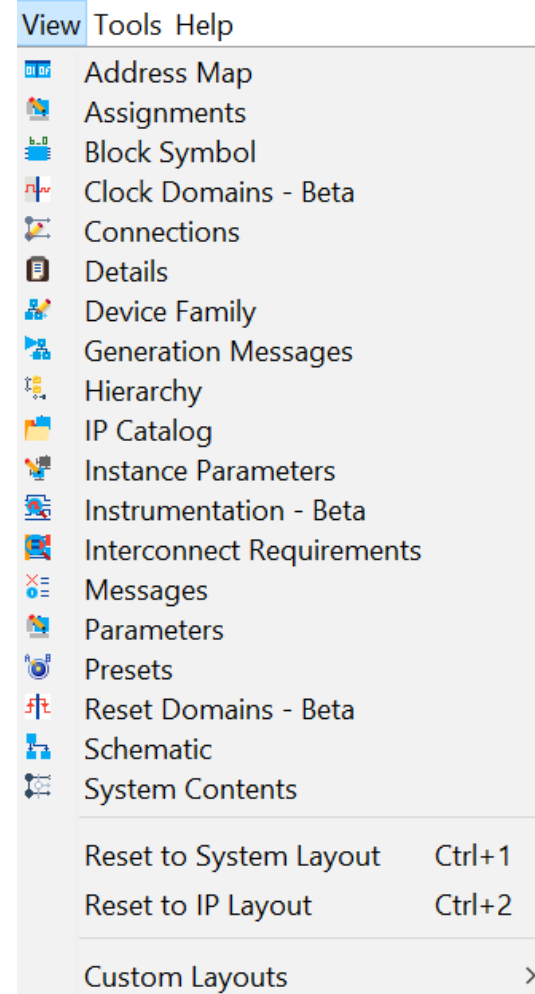
1 Error, 1 Warning

Generate HDL... Finish

Меню View

Управление раскладкой пользовательского интерфейса

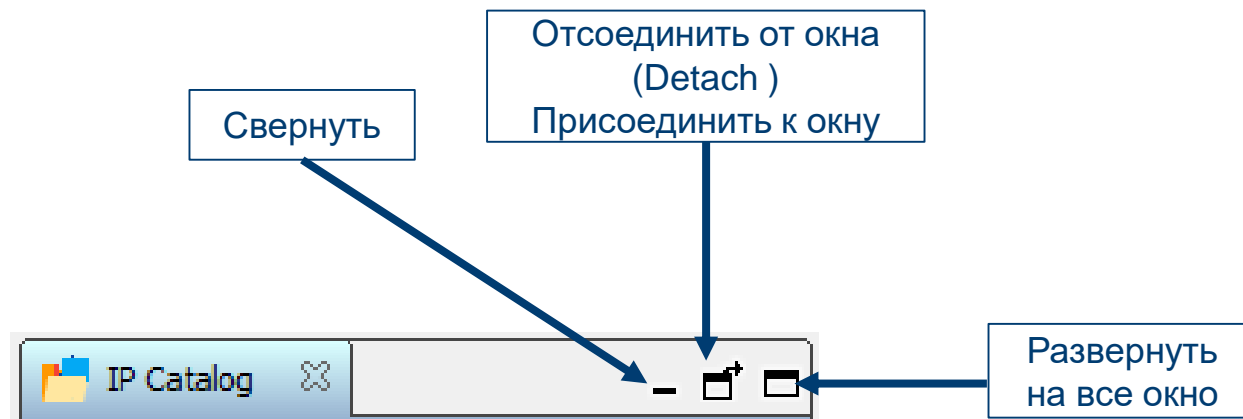
- Все элементы интерфейса организованы в виде закладок, доступных из меню View
 - Для сброса к базовому виду: меню View → Reset to System Layout
- Элементы, выбранные на одной закладке, определяют то, что отображается на других закладках
- Создание пользовательских раскладок
 - меню View → Custom layouts
 - импорт/экспорт раскладок: тип файла - .layout



Управление закладками (Tab)

Все закладки поддерживают возможность изменения положения и размера

- Закладки отсоединяются/присоединяются к окну приложения
- Закладки можно свернуть/развернуть на все окно



Библиотека IP

Библиотека проекта

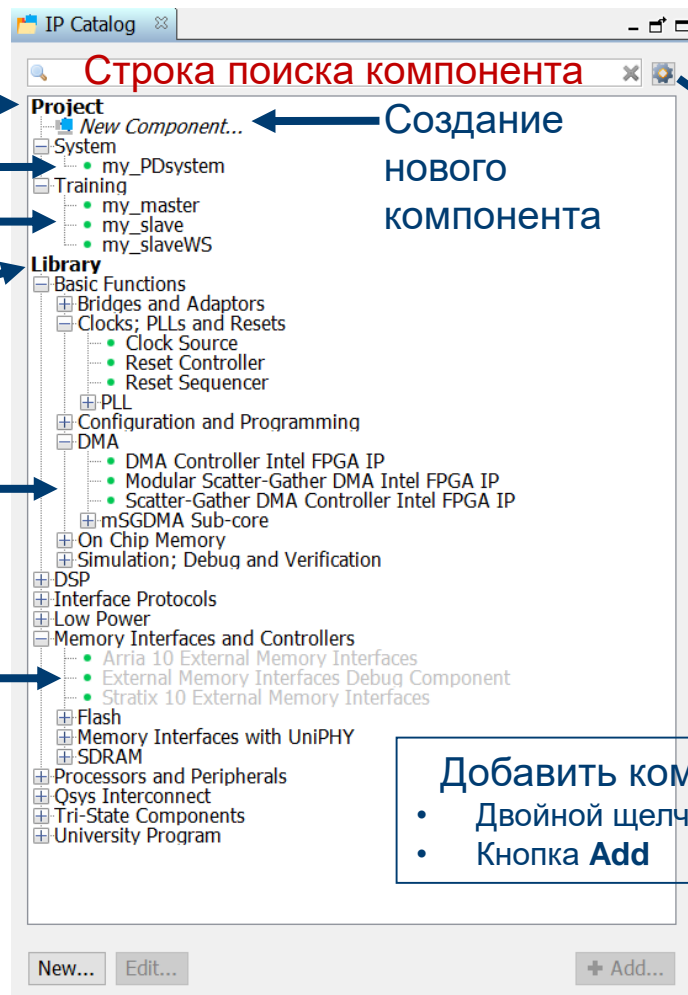
Библиотека созданных систем проекта

Пользовательские компоненты проекта

Библиотека пакета QP

Компоненты, которые
МОЖНО использовать

Компоненты, которые
НЕЛЬЗЯ использовать



Добавить компонент к системе:

- Двойной щелчок или
- Кнопка Add

Закладка структура системы (System Contents Tab)

Компоненты и подсистемы

Адреса

Project

- New Component...
- System
 - my_hierPDsystem
 - my_PDsystem
- Training
 - my_master
 - my_slave
 - my_slaveWS

Library

- Basic Functions
- DSP
- Interface Protocols
- Low Power
- Memory Interfaces and Controllers
- Processors and Peripherals
- Qsys Interconnect
- Tri-State Components
- University Program

System: my_hierPDsystem Path: PDsystem_0

Use	Connect	Name	Description	Export	Clock	Base	End
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	clk_0	Clock Source	clk	exported		
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	clk_in	Clock Input	reset	clk_0		
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	clk_in_reset	Reset Input	clk			
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	clk	Clock Output	clk_reset			
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	clk_reset	Reset Output				
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	PDsystem_0	my_PDsystem				
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	clk	Clock Input	dout_a	clk_0		
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	dout_a	Conduit	dout_b	[clk]		
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	dout_b	Conduit	reset	[clk]		
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	reset	Reset Input				
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	my_master_0	my_master				
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	clock	Clock Input	dout_a	clk_0		
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	reset	Reset Input	dout_b	[clock]		
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	m0	Avalon Memory Mapped ...		[clock]		
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	my_slave_0	my_slave				
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	clock	Clock Input	dout2	clk_0	0x0000_0004	0x0000_0004
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	reset	Reset Input		[clock]		
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	s0	Avalon Memory Mapped ...		[clock]		
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	conduit_end_0	Conduit		[clock]		
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	my_slaveWS_0	my_slaveWS				
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	clock	Clock Input	dout1	clk_0	0x0000_0000	0x0000_0003
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	reset	Reset Input		[clock]		
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	s0	Avalon Memory Mapped ...		[clock]		
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	conduit_end_0	Conduit		[clock]		

Current filter:

Подключения

Внешние выводы

Закладка структура системы (System Contents Tab)

■ Инструменты управления



Добавить компонент



Добавить подсистему



Удалить компонент



Настроить компонент



Переместить вверх



Переместить вверх на один шаг



Переместить вниз на один шаг



Переместить вниз

System Contents Address Map Interconnect Requirements

System: sm_transfer_system Path: clk

Use Connections

Компоненты

Подключить/отключить

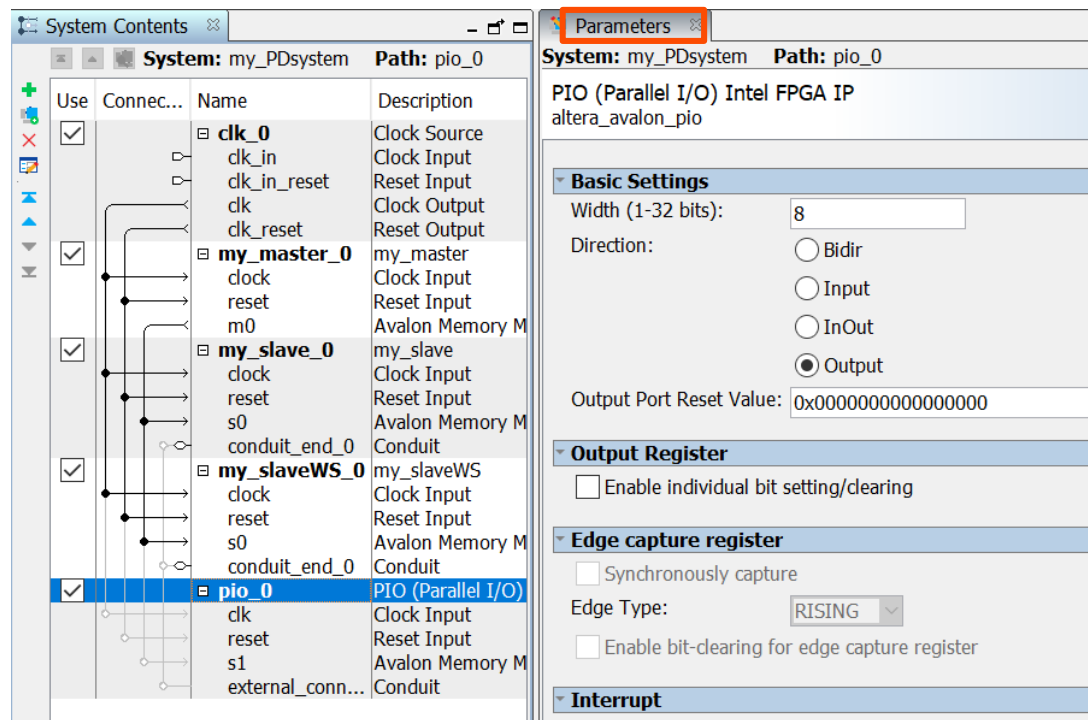
Удалить компонент – выделить и нажать **Delete**

Name	Description
clk	Clock Source
clk_in	Clock Input
clk_in_reset	Reset Input
clk	Clock Output
clk_reset	Reset Output
pll	Altera PLL
refclk	Clock Input
reset	Reset Input
outclk0	Clock Output
outclk1	Clock Output
locked	Conduit
reset_debounce	Reset Button Debounce
clock	Clock Input
button_debounce_in	Reset Input
button_qual	Reset Output
button_qual_n	Reset Output
button_switch	Button Switch PIO
clock	Clock Input
reset	Reset Input
buttonreg	Avalon Memory Mapped Slave
button_conduit	Conduit
switch_conduit	Conduit
av_sm_master	Avalon State Machine Master
clock	Clock Input
reset	Reset Input
avalon_master	Avalon Memory Mapped Master

Закладка Параметры (Parameter Tab)

При добавлении компонента к структуре системы (System Contents tab):

- Компонент помещается под всеми добавленными ранее компонентами
- Открывается окно настройки параметров компонента
- Как можно настроить ранее добавленные компоненты:
 - Двойным щелчком по имени компонента - открывается закладка настройки параметров
 - Или щелкнуть правой клавишей мыши и выбрать команду Edit



Интерфейсы компонента

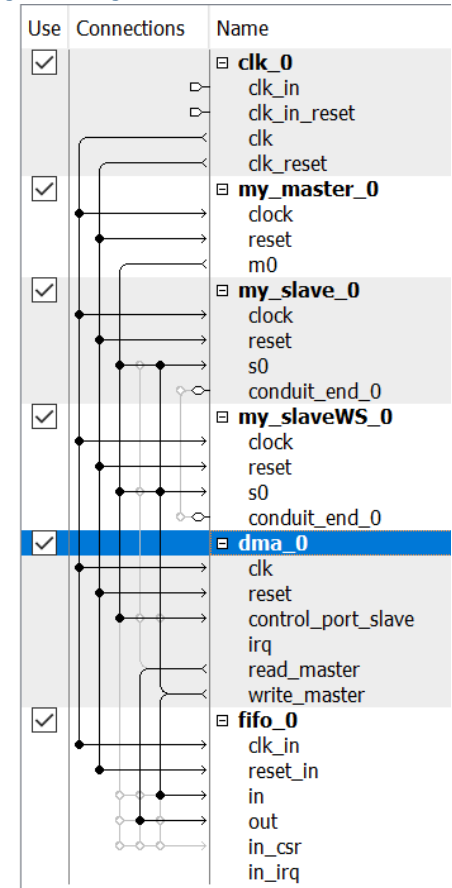
- Интерфейс: группа из одного или нескольких сигналов, которые могут быть подключены к другим интерфейсам в системе
- Ссылка на интерфейс
 - Пример: `source.s1` интерфейс `s1` (Avalon MM slave) компонента `source`
- Сигналы `clk` и `reset` – два отдельных интерфейса

	[-] dma_source_to_ssr...	DMA Controller
→	clk	Clock Input
→	reset	Reset Input
→	control_port_slave	Avalon Memory Mapped Slave
	irq	Interrupt Sender
←	read_master	Avalon Memory Mapped Master
←	write_master	Avalon Memory Mapped Master
	[-] source	On-Chip Memory (RAM or ROM)
→	clk1	Clock Input
→	s1	Avalon Memory Mapped Slave
→	reset1	Reset Input

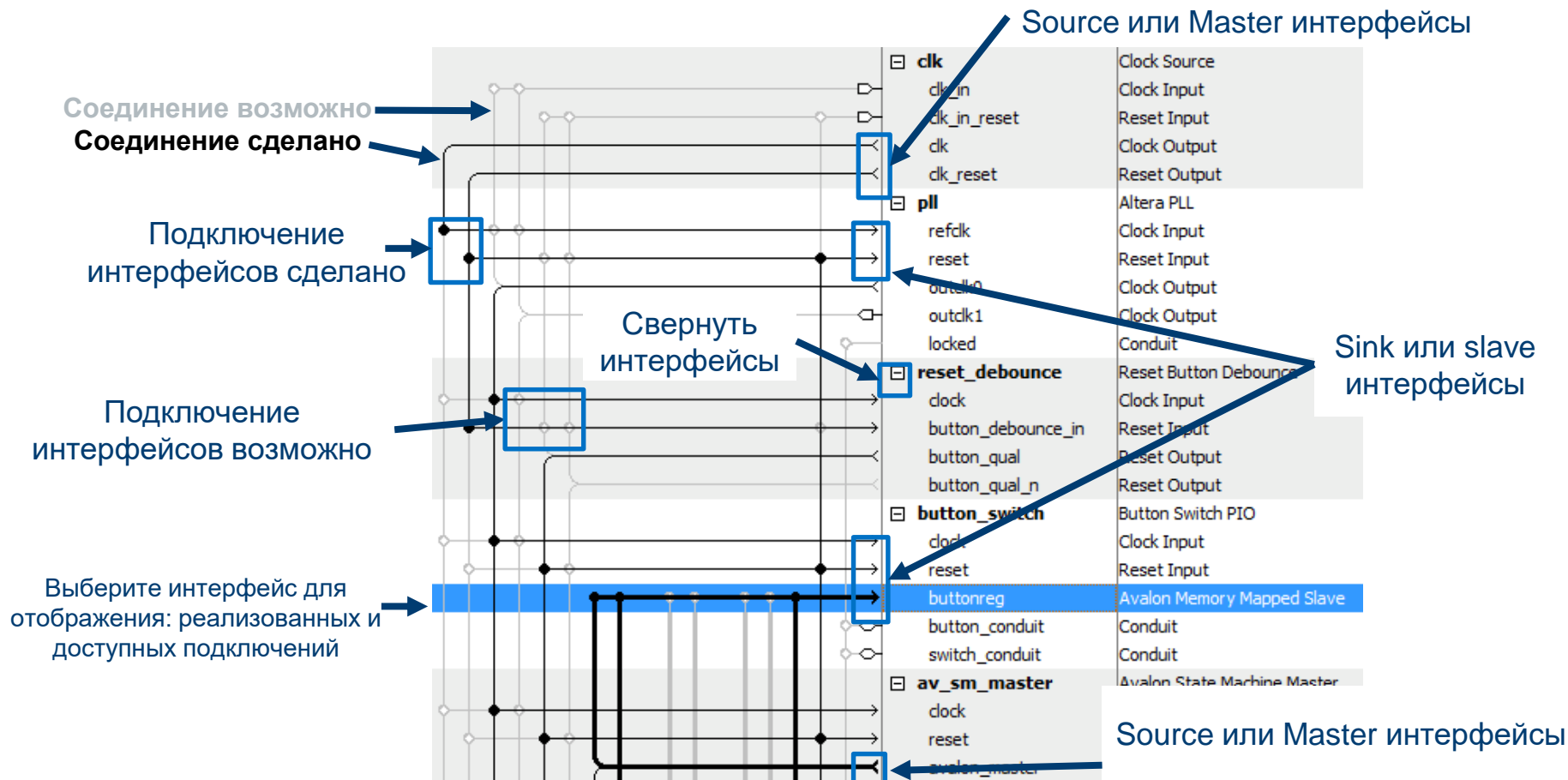
Панель коммутации Connections (Структура системы)

Колонка Connections – коммутационная панель для соединения компонентов на закладке Структура Системы

- Позволяет соединять только совместимые интерфейсы
 - Clock и reset источники (sources) => к входам компонентов
 - Masters => к slaves (для Avalon MM)
 - Sources => к sinks (для Avalon ST)
- Черные точки – реализованные подключения
- Серые точки – допустимые подключения
- PD генерирует систему связей на основе выполненных подключений

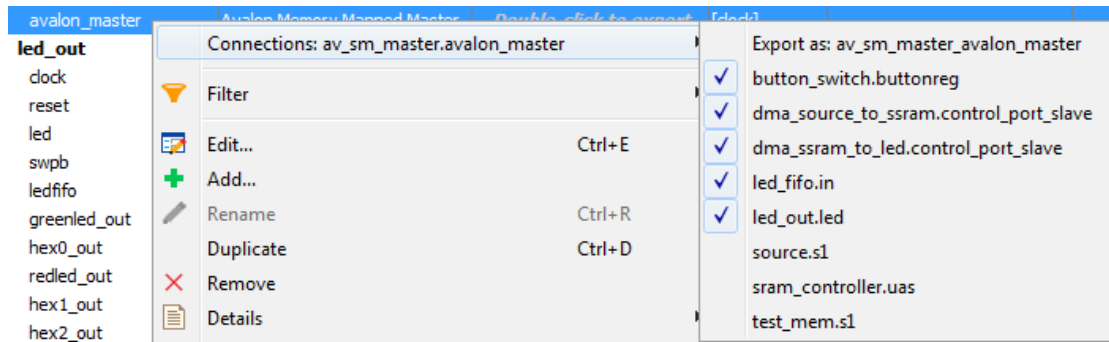


Особенности Панели коммутации (Connections)

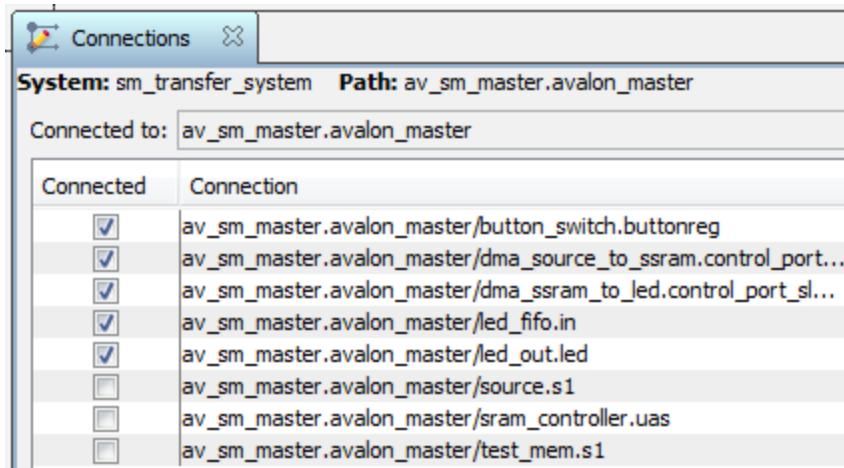


Другие способы подключения интерфейсов

Right-click на любом
интерфейсе →
Connections



Выберите интерфейс
в окне структуры
системы (**System Contents**).
Затем выполните команду
меню **View** → **Connections**



Экспортирование интерфейса

- Позволяет подключить интерфейсы к компонентам/элементам вне системы, создаваемой в PD
 - Любой интерфейс может быть экспортирован
 - Интерфейс может быть экспортирован ИЛИ подключен в системе
- Экспорт интерфейса необходим для подключения к выводам FPGA

Экспортированный интерфейс отображается как ВЫВОД

Name	Description	Export
clk	Clock Source	clk
clk_in	Clock Input	reset
clk_in_reset	Reset Input	<i>Double-click to export</i>
clk	Clock Output	<i>Double-click to export</i>
clk_reset	Reset Output	<i>Double-click to export</i>
pll	Altera PLL	
refclk	Clock Input	<i>Double-click to export</i>
reset	Reset Input	<i>Double-click to export</i>
outclk0	Clock Output	<i>Double-click to export</i>
outclk1	Clock Output	<i>Double-click to export</i>
locked	Conduit	
reset_debounce	Reset Button Debounce	ssram
clock	Clock Input	<i>Double-click to export</i>
button_debounce_in	Reset Input	<i>Double-click to export</i>
button_qual	Reset Output	<i>Double-click to export</i>
button_qual_n	Reset Output	<i>Double-click to export</i>
button_switch	Button Switch PIO	
clock	Clock Input	<i>Double-click to export</i>
reset	Reset Input	<i>Double-click to export</i>
buttonreg	Avalon Memory Mapped Slave	<i>Double-click to export</i>
button_conduit	Conduit	buttons
switch_conduit	Conduit	switches

- Как экспортировать: Double-click и ввести имя
- Как убрать экспортирование: удалить имя

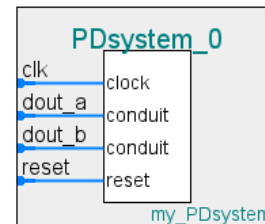
Тактовый сигнал (Clock)

Все операции в системе синхронны, поэтому необходим тактовый сигнал

- Каждый новый .qsys файл (новая система) по умолчанию включает компонент Clock Source
 - Этот компонент опциональный и может быть удален
 - Компонент соединяет два интерфейса
 - Экспортированный интерфейс Clock, приходящий из вне проектируемой системы (Clock Input)
 - Source интерфейс (Clock Output), подключаемый внутри системы к тактовым входам (sink)

КОМПОНЕНТОВ

Use	Connec...	Name	Description	Export	Clock
<input checked="" type="checkbox"/>		clk_0	Clock Source		
		clk_in	Clock Input	clk	exported
		clk_in_reset	Reset Input	reset	
		clk	Clock Output	Double-click to	clk_0
		clk_reset	Reset Output	Double-click to	
<input checked="" type="checkbox"/>		my_master_0	my_master		
		clock	Clock Input	Double-click to	clk_0
		reset	Reset Input	Double-click to	[clock]
		m0	Avalon Memory Mapped Master	Double-click to	[clock]
<input checked="" type="checkbox"/>		my_slave_0	my_slave		
		clock	Clock Input	Double-click to	clk_0
		reset	Reset Input	Double-click to	[clock]
		s0	Avalon Memory Mapped Slave	Double-click to	[clock]
		conduit_end_0	Conduit	dout_a	[clock]



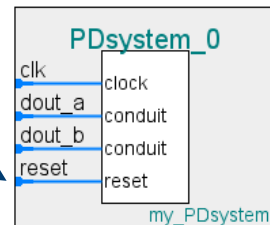
Колонка Clock упрощает подключение к тактовым сигналам

Управление сбросом системы (Reset)

PD позволяет осуществлять управление сбросом системы

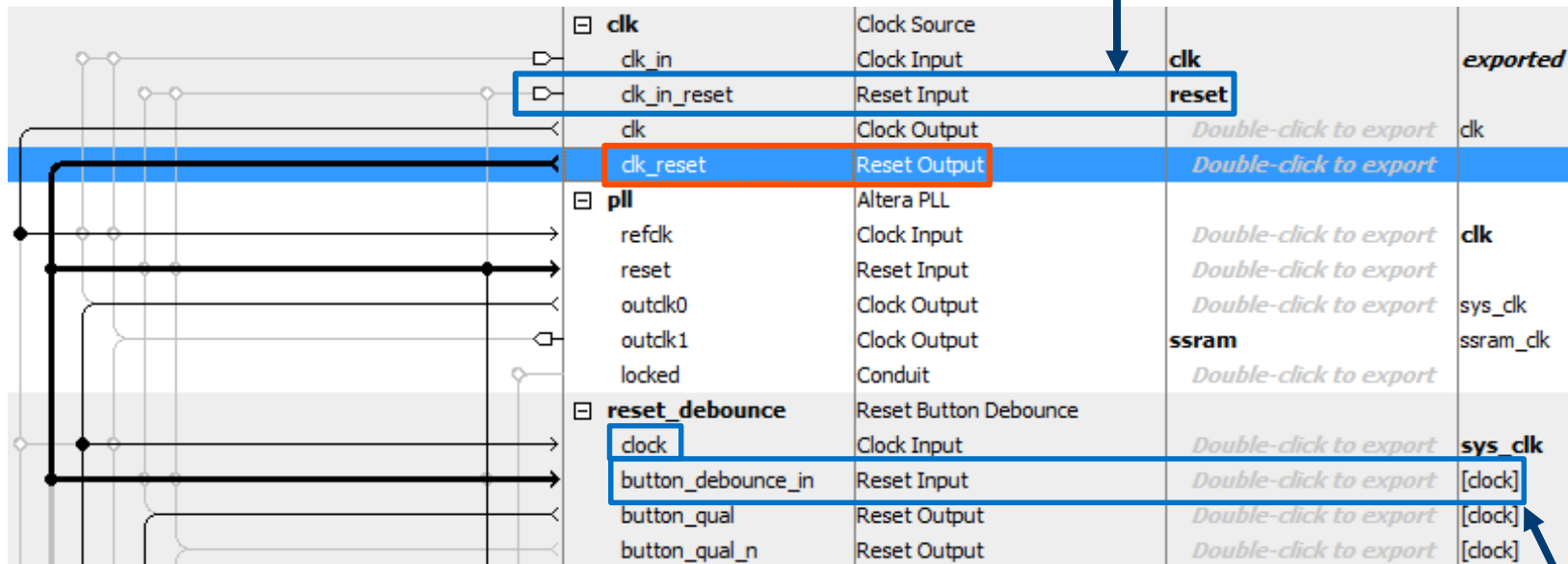
- Можно сбрасывать всю систему или отдельные компоненты системы
- Сигнал Reset – отдельный интерфейс
 - Вход (sink) привязан к clock интерфейсу и синхронизируется им
- Система может содержать несколько сигналов сброса
- IP модули Reset Controller и Reset Sequencer позволяют управлять сбросами системы

Use	Connec...	Name	Description	Export	Clock
<input checked="" type="checkbox"/>		clk_0	Clock Source		
		clk in	Clock Input	clk	exported
		clk in reset	Reset Input	reset	
		clk	Clock Output	Double-click to	clk_0
		clk_reset	Reset Output	Double-click to	
<input checked="" type="checkbox"/>		my_master_0	my_master		
		clock	Clock Input	Double-click to	clk_0
		reset	Reset Input	Double-click to	[clock]
		m0	Avalon Memory Mapped Master	Double-click to	[clock]
<input checked="" type="checkbox"/>		my_slave_0	my_slave		
		clock	Clock Input	Double-click to	clk_0
		reset	Reset Input	Double-click to	[clock]
		s0	Avalon Memory Mapped Slave	Double-click to	[clock]
		conduit_end_0	Conduit	dout_a	[clock]



Подключение сигнала сброса (Reset)

Компонент **Clock Source** подключает
внешний сигнал Сброса (Reset) – экспортированный вход



Автоматическое подключение сигналов сброса
Меню **System** → **Create Global Reset Network**

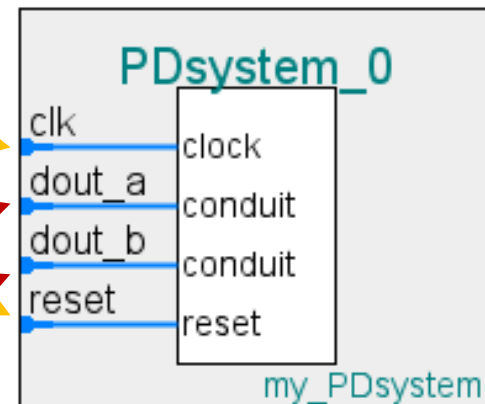
Вход Reset синхронизируется
сигналом **clock**

Интерфейс Conduit

Компоненты используют интерфейс Conduit для сигналов, которые не соответствуют стандартным интерфейсам

- Примеры стандартных интерфейсов: Avalon_MM, Avalon_ST, Arm* AXI
- В пользовательских компонентах разработчик должен самостоятельно определить сигналы интерфейса Conduit
- Чаще всего интерфейс Conduit используется для экспортирования выводов.

Use	Connec...	Name	Description	Export	Clock
<input checked="" type="checkbox"/>		clk_0	Clock Source		
		clk_in	Clock Input	clk	
		clk_in_reset	Reset Input	reset	
		clk	Clock Output	<i>Double-click to</i>	<i>exported</i>
		clk_reset	Reset Output	<i>Double-click to</i>	clk_0
<input checked="" type="checkbox"/>		my_master_0	my_master		
		clock	Clock Input	<i>Double-click to</i>	clk_0
		reset	Reset Input	<i>Double-click to</i>	[clock]
		m0	Avalon Memory Mapped Master	<i>Double-click to</i>	[clock]
<input checked="" type="checkbox"/>		my_slave_0	my_slave		
		clock	Clock Input	<i>Double-click to</i>	clk_0
		reset	Reset Input	<i>Double-click to</i>	[clock]
		s0	Avalon Memory Mapped Slave	<i>Double-click to</i>	[clock]
		conduit_end_0	Conduit	<i>Double-click to</i>	dout_a
<input checked="" type="checkbox"/>		my_slaveWS_0	my_slaveWS		
		clock	Clock Input	<i>Double-click to</i>	clk_0
		reset	Reset Input	<i>Double-click to</i>	[clock]
		s0	Avalon Memory Mapped Slave	<i>Double-click to</i>	[clock]
		conduit_end_0	Conduit	<i>Double-click to</i>	dout_b



Адресация Memory-Mapped (MM)

Каждый MM master интерфейс имеет собственное адресное пространство

- Максимальный размер адресного пространства определяется разрядностью адреса - 64
- Карта памяти (Memory Map) каждого MM Master интерфейса формируется:
 - Базовым адресом подключенных ведомых (slave)
 - Диапазоном адресов (address spans) подключенных ведомых (slave)
 - *Диапазоны адресов подключенных ведомых (slave) не должны пересекаться*
- Не перекрывающиеся базовые адреса назначаются:
 - Либо вручную; либо командой: меню System → Assign Base Addresses

Master интерфейс		Зафиксировать адрес		Карта памяти (0x0 to 0x6f)	
⊕ av_sm_master	Avalon State Machine Master	sys_clk			
⊕ led_fifo	On-Chip FIFO Memory	sys_clk	🔒	0x00000000	0x00000007
⊕ dma_ssramm_to_led	DMA Controller	sys_clk	🔒	0x00000020	0x0000003f
⊕ dma_source_to_ssram	DMA Controller	sys_clk	🔒	0x00000040	0x0000005f
⊕ start_pushbutton	PIO (Parallel I/O)	sys_clk	🔒	0x00000060	0x0000006f

Закладка Карта Памяти (Address Map Tab)

Закладка Address Map tab - средство анализа и управление адресацией:

- Отображает адреса для реализованных подключений MM Master => MM Slave
- Для редактирования ячейки - Double-click ячейку
- Поддерживает адресацию для разделяемых ведомых (shared slaves)
 - Разделяемый ведомый может иметь разные адреса для подключённых ведущих

System Contents Address Map Interconnect Requirements

System: sm_transfer_system Path: test_mem

	J2A_master.master	av_sm_master.avalon_master	dma_ssram_to_led.write_master
button_switch.buttonreg	0x0000_1040 - 0x0000_104f	0x0000_1040 - 0x0000_104f	
dma_source_to_ssram.control_port_sl...		0x0000_0040 - 0x0000_005f	
dma_ssram_to_led.control_port_slave		0x0000_0020 - 0x0000_003f	
led_fifo.in	0x0000_1050 - 0x0000_1057	0x0000_0000 - 0x0000_0007	0x0000_0000 - 0x0000_0007
led_out.led	0x0000_1030 - 0x0000_103f	0x0000_1030 - 0x0000_103f	
source.s1			
ssram_controller.uas			
test_mem.s1			0x0010_00

Slave интерфейсы

Master интерфейсы

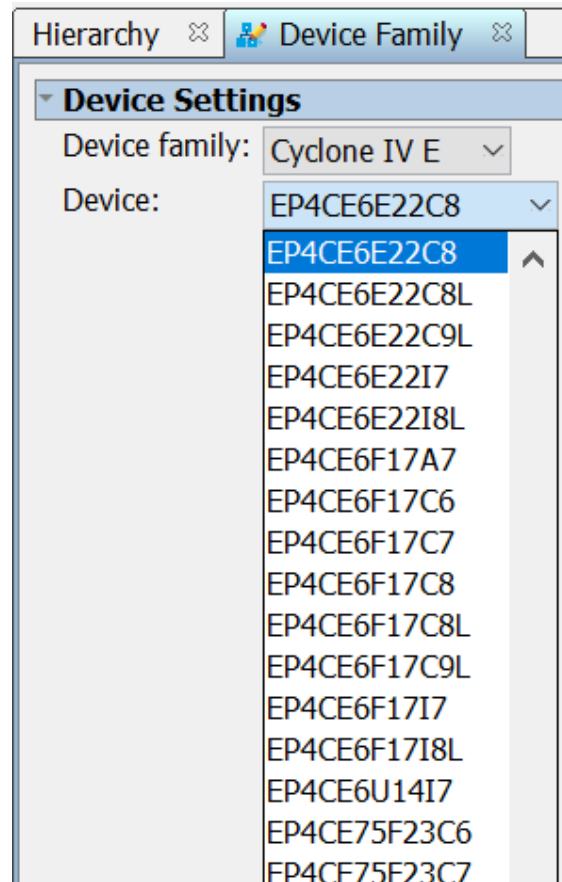
led_fifo

Signal	Interface	Properties	Mode
clk_in	Avalon FIFO Memory		
reset_in	Clock Input		
in	Reset Input		
in	Avalon Memory Mapped Slave		
out	Avalon Streaming Source	mixed	mixed

Закладка Семейство микросхем (Device Family)

Закладка позволяет:

- Отобразить назначенное семейство и тип микросхемы FPGA
- Изменить\задать семейство и тип микросхемы FPGA



Закладка Interconnect Requirements Tabs

Позволяет задать общие требования к системе межсоединений

Установить число уровней конвейеризации

Установка дополнительных требований и инструментов Intel Quartus Prime software handbook описывает детали

System Contents Address Map **Interconnect Requirements**

Configure interconnect requirements for the system or an interface.

System-wide Requirements

Limit interconnect pipeline stages to: 2

Clock crossing adapter type: Handshake

All Requirements

Identifier	Setting	Value
\$system	Clock crossing adapter type	Handshake
\$system	Limit interconnect pipeline stages to	2
\$system	Enable instrumentation	TRUE
av_sm_master.avalon_master	Add performance monitor	TRUE
button_switch.buttonreg	Add performance monitor	TRUE
button_switch.buttonreg	Security	Non-secure
led_fifo.in	Add performance monitor	TRUE
led_out.led	Add performance monitor	TRUE
led_out.swpb	Add performance monitor	TRUE
source.s1	Add performance monitor	TRUE

Security

Secure address ranges

Multiple threadID support

ThreadID buffer depth

ThreadID mapping ranges

Enable master reorder buffer

Add performance monitor

Slave arbitration scheme

Handshake, FIFO, Auto

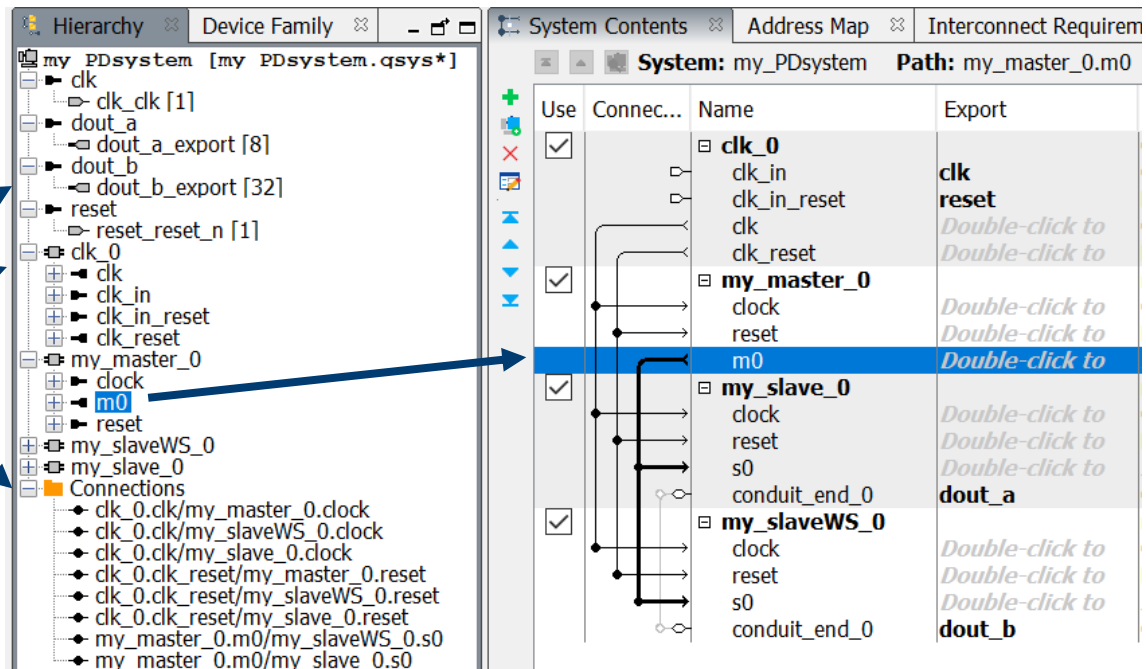
Закладка Иерархия (Hierarchy Tab)

Средство анализа системы
отображает:

- Иерархию системы
- Экспортированные интерфейсы
- Интерфейсы компонентов
- Связи между компонентами

Позволяет:

- Редактировать настройки (Edit)
- Отображать выбранные элементы в других закладках (Cross-highlight)



Закладка Символ (Block Symbol)

Выбранный уровень
иерархии системы

Позволяет отобразить все
сигналы интерфейсов

The screenshot displays the 'Block Symbol' tab in a design tool. The left pane shows the hierarchy of the system 'my_PDsystem'. The main pane shows the 'System: my_PDsystem' with a table of components and their connections. The right pane shows the 'Block Symbol' view, which displays the system's external signals.

System: my_PDsystem

Use	Connec...	Name	Description	Export	Clock
<input checked="" type="checkbox"/>		clk_0	Clock Source		
		clk_in	Clock Input	clk	
		clk_in_reset	Reset Input	reset	
		clk	Clock Output	Double-click to	clk_0
		clk_reset	Reset Output	Double-click to	
<input checked="" type="checkbox"/>		my_master_0	my_master		
		clock	Clock Input	Double-click to	clk_0
		reset	Reset Input	Double-click to	[clock]
		m0	Avalon Memory Mapped Master	Double-click to	[clock]
<input checked="" type="checkbox"/>		my_slave_0	my_slave		
		clock	Clock Input	Double-click to	clk_0
		reset	Reset Input	Double-click to	[clock]
		s0	Avalon Memory Mapped Slave	Double-click to	[clock]
		conduit_end_0	Conduit	Double-click to	[clock]
<input checked="" type="checkbox"/>		my_slaveWS_0	my_slaveWS		
		clock	Clock Input	Double-click to	clk_0
		reset	Reset Input	Double-click to	[clock]
		s0	Avalon Memory Mapped Slave	Double-click to	[clock]
		conduit_end_0	Conduit	Double-click to	[clock]

Block Symbol

☒ Show signals

The Block Symbol view shows the system's external signals:

- clk_clk
- clk
- dout_a_export[7..0]
- dout_b_export[31..0]
- reset
- reset_reset_n

Символ отображает все внешние (экспортированные) интерфейсы системы

Закладка Схема (Schematic View)

Отображение структуры системы и управление настройками

my_PDsystem [my_PDsystem.qsys*]

clk

dout_a

dout_b

reset

clk_0

my_master_0

clk

m0

reset

my_slaveWS_0

my_slave_0

clk

conduit_end_0

reset

s0

Connections

clk_0.clk/my_master_0.clk

clk_0.clk/my_slaveWS_0.clk

clk_0.clk/my_slave_0.clk

clk_0.clk_reset/my_master_0.reset

clk_0.clk_reset/my_slaveWS_0.reset

clk_0.clk_reset/my_slave_0.reset

my_master_0.m0/my_slaveWS_0.s0

my_master_0.m0/my_slave_0.s0

System Contents

Address Map

Interconnect Requirements

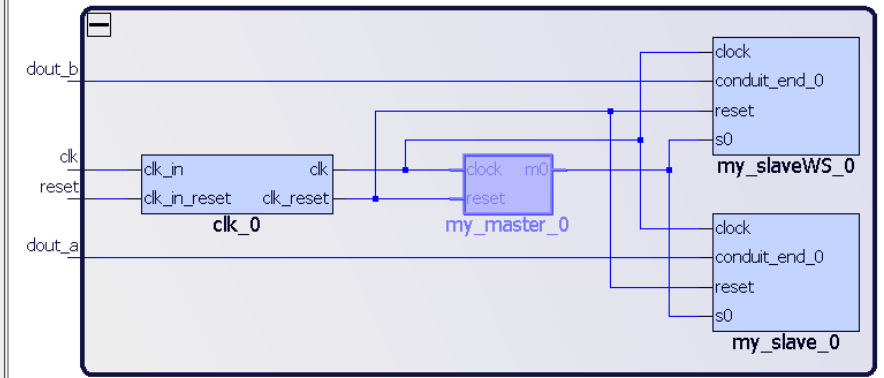
Schematic

Block Symbol

Parameters

System: my_PDsystem

Path: my_master_0



my_master_0

clock

csi_clk

reset

rsi_reset

clk

address

write

writedata

waitrequest

avm_m0_address[31..0]

avm_m0_writedata[31..0]

avm_m0_waitrequest

my_master

Сохранить .pdf

Zoom in/out

Exclude: rst clk st mm

Enter a comma-separated list of instances or types to exclude.

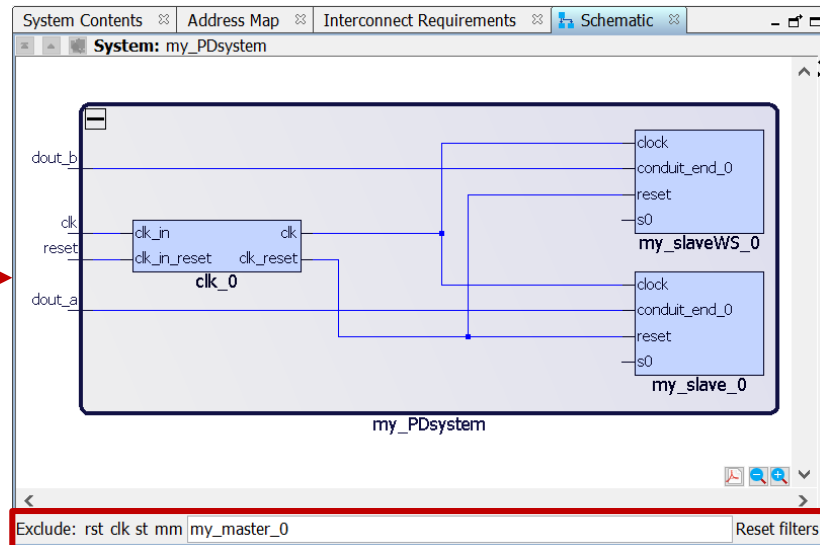
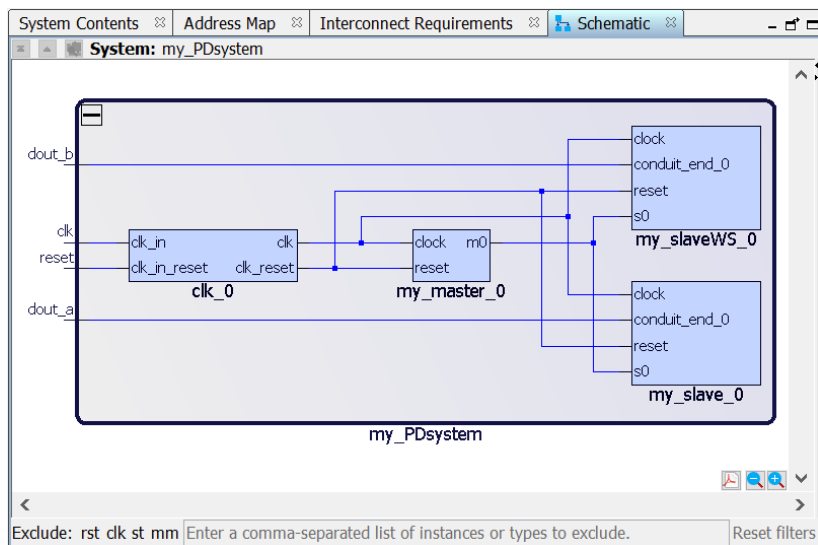
Reset filters

Отображение
выбранных
элементов в
других закладках
(Cross-highlight)

Фильтрация отображаемых соединений

Закладка Схема (Schematic View)

Фильтрация отображаемых элементов



Фильтрация отображаемых соединений
и элементов

Сброс
фильтра

Закладка Сообщения (Messages Tab)

Закладка позволяет:

- Отображать сообщения (ошибки, предупреждения, информацию)
- Отображать источник ошибки/предупреждения - Double-click сообщение для выделения интерфейса/соединения или компонента
- Не должно быть ошибок для генерации системы
 - В процессе настройки ошибки могут (и будут) присутствовать

Type	Path	Message
2 Errors		
Error	my_PDSystem.my_master_0.m0	my_slaveWS_0.s0 cannot be at 0x2 (0x0 or 0x4 are acceptable)
Error	my_PDSystem.my_slaveWS_0	my_slaveWS_0.clock must be connected to a clock output
1 Warning		
Warning	my_PDSystem.my_slave_0	my_slave_0.conduit_end_0 must be exported, or connected to a matching conduit.

Сообщения
(info, warning, error)

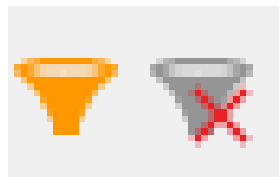
Использование фильтра (System Contents Tab)

Управление отображением системы в закладке System Contents

- Существуют predefined фильтры и можно создавать пользовательские фильтры

The screenshot shows the 'System Contents' window for a system named 'my_PDsystem'. A green box highlights the 'Use' column, which contains a tree view of components. Another green box highlights the 'Filters' dialog box, which is open. The dialog shows the 'Filter' dropdown set to 'Clock and Reset Interfaces'. Below this, there are sections for 'match any' and 'and match all' with tables for defining rules. The 'match any' section has two rules: 'Interface type is clock' and 'Interface type is reset'. The 'and match all' section is empty. At the bottom, a status bar indicates 'Current filter: Clock and Reset Interfaces'.

Co...	Name	Description	Export	Clock
<input checked="" type="checkbox"/>	clk_0	Clock Source	clk	exported
<input checked="" type="checkbox"/>	clk_in	Clock Input	reset	[clk_in]
<input checked="" type="checkbox"/>	clk_in_reset	Reset Input	Double-click to	clk_0
<input checked="" type="checkbox"/>	clk	Clock Output	Double-click to	clk_0
<input checked="" type="checkbox"/>	clk_reset	Reset Output	Double-click to	clk_0
<input checked="" type="checkbox"/>	my_master_0	my_master	Double-click to	clk_0
<input checked="" type="checkbox"/>	clock	Clock Input	Double-click to	clk_0
<input checked="" type="checkbox"/>	reset	Reset Input	Double-click to	clk_0
<input checked="" type="checkbox"/>	my_slave_0	my_slave	Double-click to	clk_0
<input checked="" type="checkbox"/>	clock	Clock Input	Double-click to	clk_0
<input checked="" type="checkbox"/>	reset	Reset Input	Double-click to	clk_0
<input checked="" type="checkbox"/>	my_slaveWS_0	my_slaveWS	Double-click to	clk_0
<input checked="" type="checkbox"/>	clock	Clock Input	Double-click to	clk_0
<input checked="" type="checkbox"/>	reset	Reset Input	Double-click to	clk_0



установить и сбросить
фильтр

The screenshot shows the 'System Contents' window for a system named 'my_PDsystem'. A green box highlights the 'Use' column, which contains a tree view of components. Another green box highlights the 'Filters' dialog box, which is open. The dialog shows the 'Filter' dropdown set to 'Avalon-MM Interfaces'. Below this, there are sections for 'match any' and 'and match all' with tables for defining rules. The 'match any' section has one rule: 'Interface type is avalon'. The 'and match all' section is empty. At the bottom, a status bar indicates 'Current filter: Avalon-MM Interfaces'.

Use	Name	Description	Export	Clock
<input checked="" type="checkbox"/>	my_master_0	my_master	Double-click to	[clock]
<input checked="" type="checkbox"/>	m0	Avalon Memory Mapped Master	Double-click to	clk_0
<input checked="" type="checkbox"/>	my_slave_0	my_slave	Double-click to	[clock]
<input checked="" type="checkbox"/>	s0	Avalon Memory Mapped Slave	Double-click to	clk_0
<input checked="" type="checkbox"/>	my_slaveWS_0	my_slaveWS	Double-click to	[clock]
<input checked="" type="checkbox"/>	s0	Avalon Memory Mapped Slave	Double-click to	clk_0

Отображение тактовых доменов (Clock Domains)

Отображение компонентов и интерфейсов выбранного тактового домена

Меню View → Clock Domains - Beta

The screenshot shows the 'Clock Domains - Beta' window for a system named 'my_PDSYSTEM'. The 'Path' is 'clk_0.clk_in'. The table lists the following components and interfaces:

Use	Connec...	Name	Description	Export	Clock
<input checked="" type="checkbox"/>		clk_0	Clock Source		
<input checked="" type="checkbox"/>		clk_in	Clock Input	clk	exported
<input checked="" type="checkbox"/>		clk_in_reset	Reset Input	reset	[clk_in]
<input checked="" type="checkbox"/>		clk	Clock Output		clk_0
<input checked="" type="checkbox"/>		clk_reset	Reset Output		clk_0
<input checked="" type="checkbox"/>		my_master_0	my_master		
<input checked="" type="checkbox"/>		clock	Clock Input		clk_0
<input checked="" type="checkbox"/>		reset	Reset Input		[clock]
<input checked="" type="checkbox"/>		m0	Avalon Memory Mapped Master		[clock]
<input checked="" type="checkbox"/>		my_slave_0	my_slave		
<input checked="" type="checkbox"/>		clock	Clock Input		clk_0
<input checked="" type="checkbox"/>		reset	Reset Input		[clock]
<input checked="" type="checkbox"/>		s0	Avalon Memory Mapped Slave		[clock]
<input checked="" type="checkbox"/>		conduit_end_0	Conduit	dout_a	[clock]
<input checked="" type="checkbox"/>		my_slaveWS_0	my_slaveWS		
<input checked="" type="checkbox"/>		clock	Clock Input		clk_0
<input checked="" type="checkbox"/>		reset	Reset Input		[clock]
<input checked="" type="checkbox"/>		s0	Avalon Memory Mapped Slave		[clock]
<input checked="" type="checkbox"/>		conduit_end_0	Conduit	dout_b	[clock]

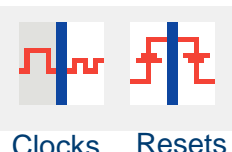
The current design has 1 clock domain
Selected clock domains:
• my_PDSYSTEM.clk - 50.0 MHz
Selected element:
• clk_0.clk_in
Export:
clk

Система с одним тактовым доменом

Система с одним тактовым доменом

Отображение интерфейсов выбранного домена

Отображение интерфейсов выбранного домена

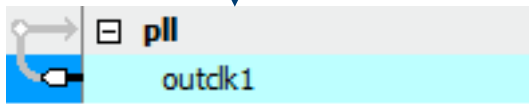
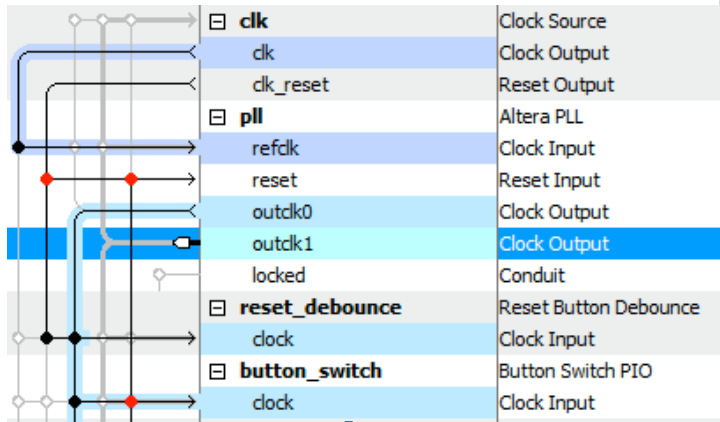
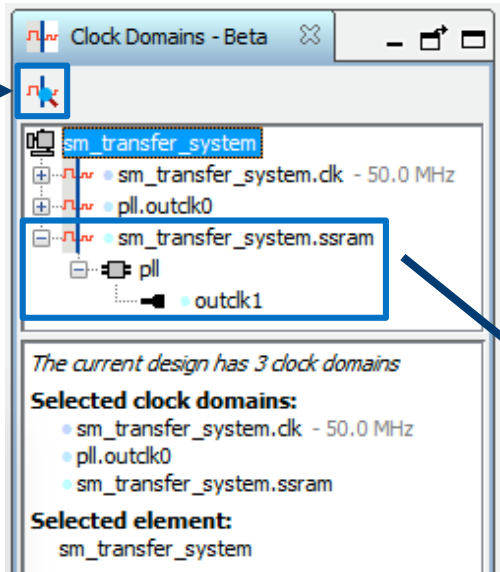


Отображение тактовых доменов (Clock Domains)

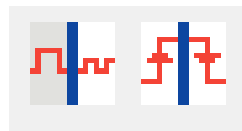
Система с несколькими тактовыми доменами

Меню View → Clock Domains - Beta

Отображение интерфейсов выбранного домена



Отображение интерфейсов выбранного домена

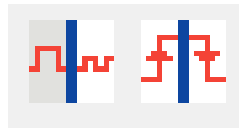


Clocks Resets



Сброс фильтра

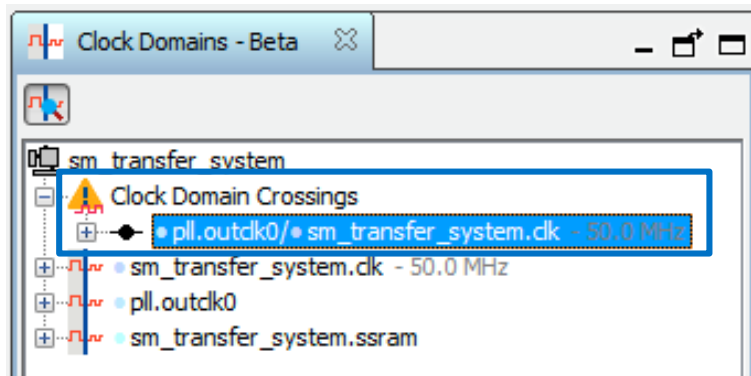
Пресечение доменов (Clock Domain Crossings)



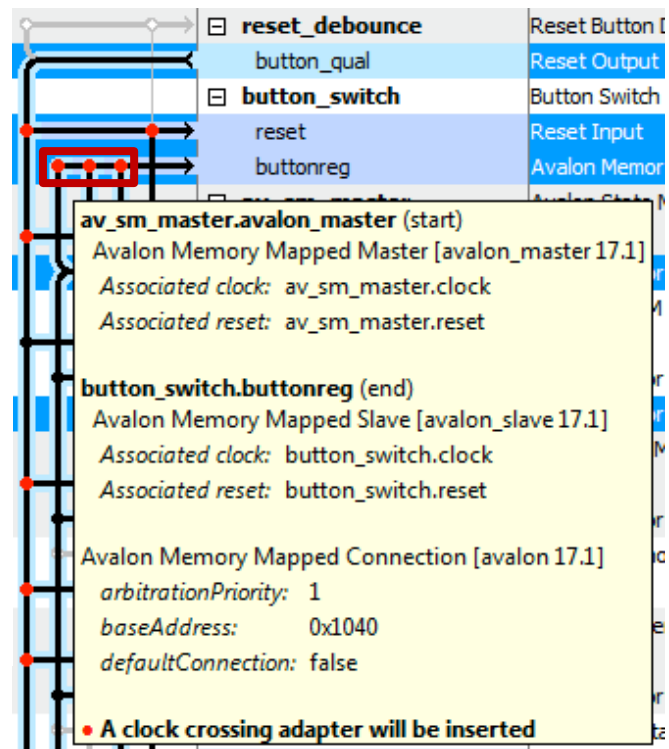
Clocks Resets

Отображаются подключения, в которых необходимо добавить адаптеры

- Места подключений отображены красными точками
- Подсказка (Tooltip) объясняет почему необходим адаптер
- Можно принять установку адаптера или изменить подключение



Ведомый (slave)
button_switch
оказался в
таковом домене
ОТЛИЧНОМ ОТ
такового домена
своего ведущего
(Master)

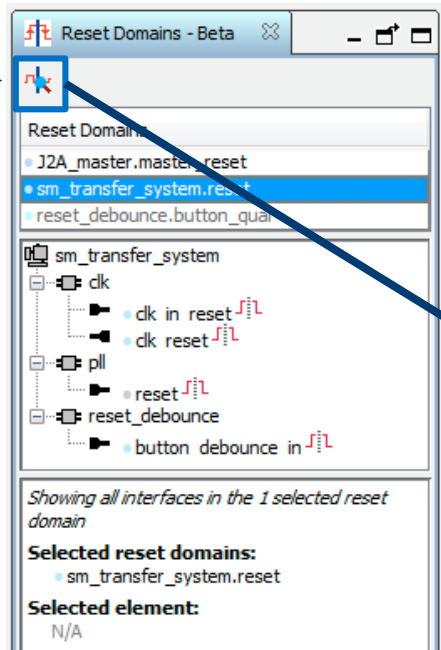


Отображение доменов Reset

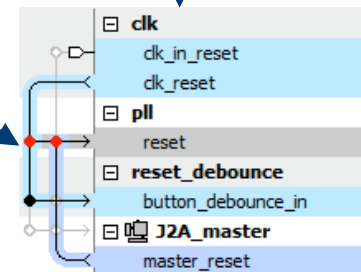
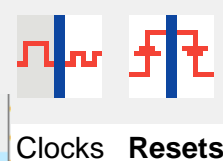
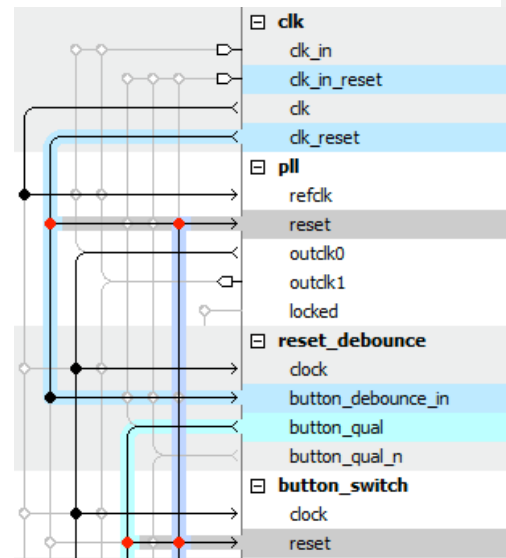
Отображение компонентов и интерфейсов выбранного домена reset

Меню **View** → **Reset Domains - Beta**

Отображение
интерфейсов
выбранного
домена



Система с несколькими тактовыми доменами



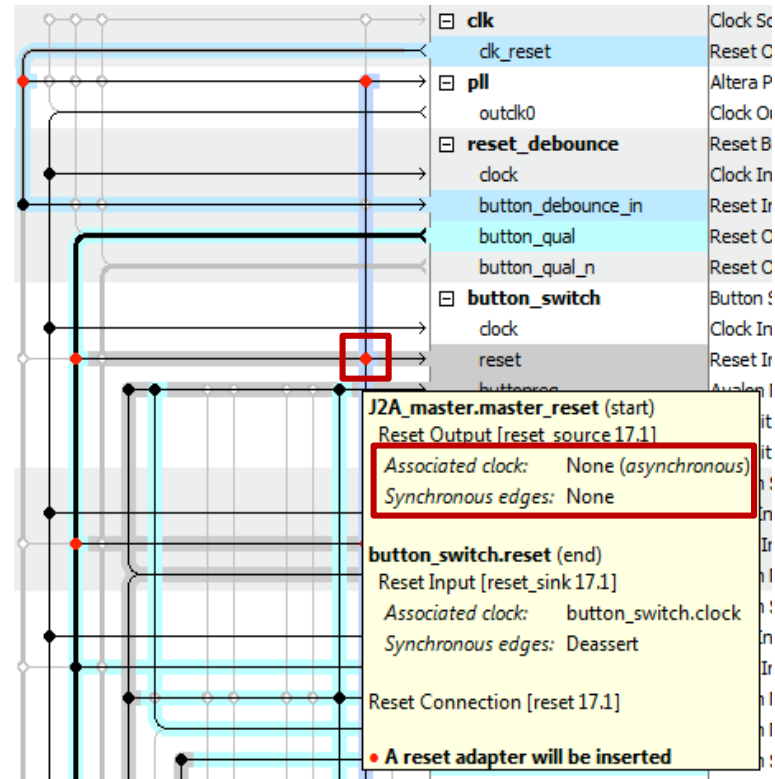
Сброс
фильтра

Пресечение доменов (Reset Domain Crossings)

Отображаются подключения, в которых необходимо добавить адаптеры

- Места подключений отображены красными точками
- Подсказка (Tooltip) объясняет почему необходим адаптер
- Можно принять установку адаптера или изменить подключение

master_reset
ассинхронный



Полезные команды

Команда: меню **System => Show System with Platform Designer Interconnect** отображает систему с модулями вставленными PD

The image shows a comparison between two views of a system in Quartus II: the 'System Contents' view on the left and the 'Memory-Mapped Interconnect' view on the right. A blue arrow points from the left view to the right view, indicating the transition.

System Contents View (Left):

Use	Connections	Name	Description	Export	Clock
<input checked="" type="checkbox"/>		clk_0 clk_in clk_in_reset clk clk_reset	Clock Source Clock Input Reset Input Clock Output Reset Output	clk reset <i>Double-click to</i> <i>Double-click to</i>	exported [clk_in] clk_0 clk_0
<input checked="" type="checkbox"/>		my_master_0 clock reset m0	my_master Clock Input Reset Input Avalon Memory Mapped Master	<i>Double-click to</i> <i>Double-click to</i>	clk_0 [clock] [clock]
<input checked="" type="checkbox"/>		my_slave_0 clock reset s0	my_slave Clock Input Reset Input Avalon Memory Mapped Slave	<i>Double-click to</i> <i>Double-click to</i> <i>Double-click to</i>	clk_0 [clock] [clock] [clock]
<input checked="" type="checkbox"/>		my_slaveWS_0 conduit_end_0 clock reset s0 conduit_end_0	my_slaveWS Conduit Clock Input Reset Input Avalon Memory Mapped Slave Conduit	<i>Double-click to</i> <i>Double-click to</i> <i>Double-click to</i> <i>Double-click to</i>	clk_0 [clock] [clock] [clock] [clock]

Memory-Mapped Interconnect View (Right):

Use	Connections	Name	Description	Export	Clock
<input checked="" type="checkbox"/>		mm_interconnect_0 clk_0_clk my_master_0_reset_reset_... my_master_0_m0 my_slave_0_s0 my_slaveWS_0_s0	MM Interconnect Clock Input Reset Input Avalon Memory Mapped Slave Avalon Memory Mapped Master Avalon Memory Mapped Master	<i>Double-click to</i> <i>Double-click to</i> <i>Double-click to</i> <i>Double-click to</i> <i>Double-click to</i>	clk_0 [clk_0_clk] [clk_0_clk] [clk_0_clk] [clk_0_clk]
<input checked="" type="checkbox"/>		rst_controller reset_in0 clk reset_out	Merlin Reset Controller Reset Input Clock Input Reset Output	<i>Double-click to</i> <i>Double-click to</i> <i>Double-click to</i>	clk_0 [clk] [clk]
<input checked="" type="checkbox"/>		clk_0 clk_in clk_in_reset clk clk_reset	Clock Source Clock Input Reset Input Clock Output Reset Output	clk reset <i>Double-click to</i> <i>Double-click to</i>	exported [clk_in] clk_0 clk_0
<input checked="" type="checkbox"/>		my_master_0 clock reset m0	my_master Clock Input Reset Input Avalon Memory Mapped Master	<i>Double-click to</i> <i>Double-click to</i> <i>Double-click to</i>	clk_0 [clock] [clock]
<input checked="" type="checkbox"/>		my_slave_0 clock reset s0 conduit_end_0	my_slave Clock Input Reset Input Avalon Memory Mapped Slave Conduit	<i>Double-click to</i> <i>Double-click to</i> <i>Double-click to</i> <i>Double-click to</i>	clk_0 [clock] [clock] [clock] [clock]
<input checked="" type="checkbox"/>		my_slaveWS_0 clock reset s0 conduit_end_0	my_slaveWS Clock Input Reset Input Avalon Memory Mapped Slave Conduit	<i>Double-click to</i> <i>Double-click to</i> <i>Double-click to</i> <i>Double-click to</i>	clk_0 [clock] [clock] [clock] [clock]

Полезные команды

- Меню **File**

- Обновить систему - Refresh System (F5)
 - Обновляет все IP и компоненты

- Меню **System**

- Обновить IP - Upgrade IP Cores
 - Если система создана в предыдущих версиях PD, то можно обновить устаревшие IP
- Назначить номера прерываний - Assign Interrupt Numbers
 - Автоматическое назначение номеров прерываний для пар interrupt sender/receiver
- Назначить код операции для пользовательской инструкции Assign Custom Instruction Opcodes
 - Используется с процессором Nios® II при создании пользовательских инструкций
- Удалить «висящие» выводы - Remove Dangling Connections
 - Удаляет неподключенные линии соединения (интерфейсы) в закладке System Contents tab

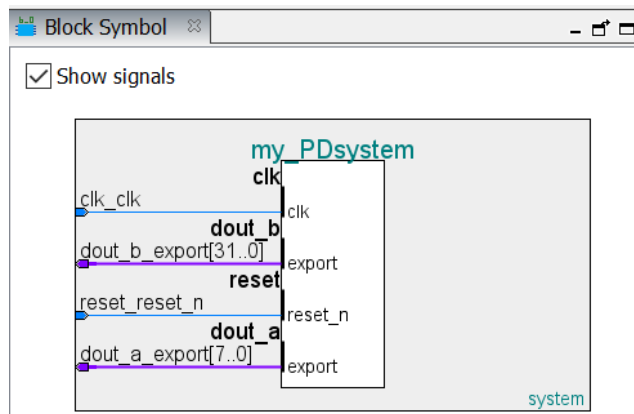
Полезные команды

- Меню **System**

- Назначить базовые адреса - Auto-Assign Base Addresses
 - Автоматически назначает допустимые базовые адреса всем ведомым (slave)
- Подключить сигналы сброса – Create Global Resets Network
 - Автоматически подключает все сигналы Reset

Заготовка HDL Instantiation Template

Заготовка для использования созданной системы как компонента в HDL описании верхнего уровня.



Меню **Generate** → **Show Instantiation Template**

Instantiation Template

You can copy the example HDL below to declare an instance of **my_PDsystem**.

HDL Language: Verilog ▾

Example HDL

```
my_PDsystem u0 (  
    .clk_clk      (<connected-to-clk_clk>),      // clk.clk  
    .dout_b_export (<connected-to-dout_b_export>), // dout_b.export  
    .reset_reset_n (<connected-to-reset_reset_n>), // reset.reset_n  
    .dout_a_export (<connected-to-dout_a_export>) // dout_a.export  
);
```

HDL Language: VHDL ▾

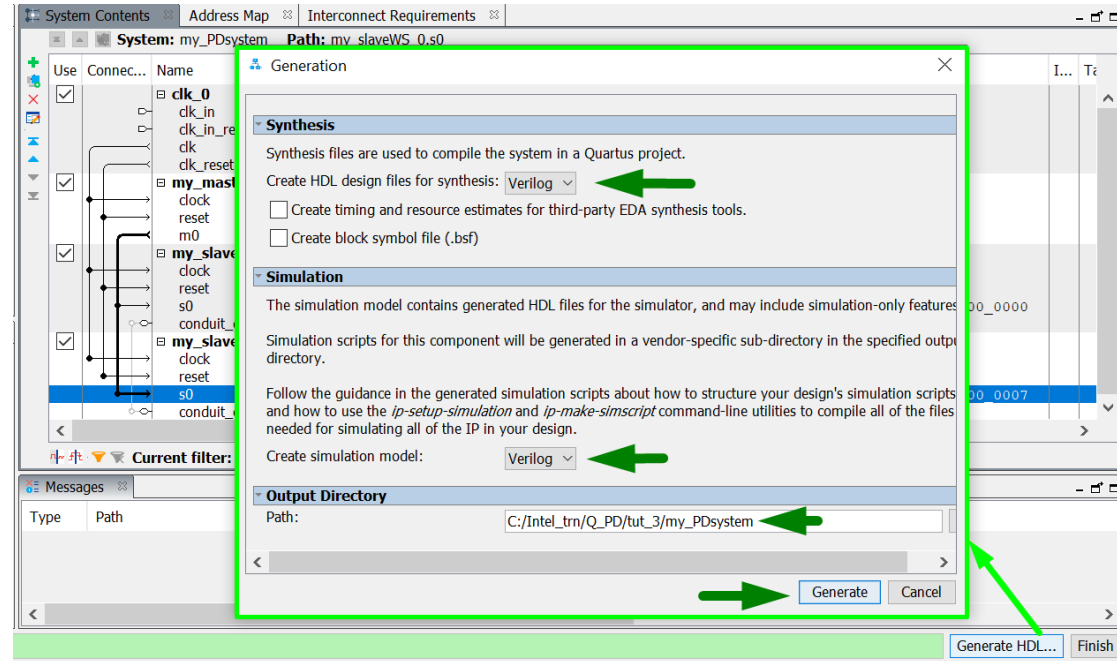
Example HDL

```
component my_PDsystem is  
    port (  
        clk_clk      : in  std_logic      := 'X'; -- clk  
        dout_b_export : out std_logic_vector(31 downto 0); -- export  
        reset_reset_n : in  std_logic      := 'X'; -- reset_n  
        dout_a_export : out std_logic_vector(7 downto 0) -- export  
    );  
end component my_PDsystem;  
  
u0 : component my_PDsystem  
    port map (  
        clk_clk      => CONNECTED_TO_clk_clk,      -- clk.clk  
        dout_b_export => CONNECTED_TO_dout_b_export, -- dout_b.export  
        reset_reset_n => CONNECTED_TO_reset_reset_n, -- reset.reset_n  
        dout_a_export => CONNECTED_TO_dout_a_export -- dout_a.export  
    );
```

Генерация HDL описания системы

Реализуется после создания системы

- Для запуска настройки процедуры используется кнопка **Generate HDL**
- Окно **Generation** позволяет:
 - Выбрать язык (Verilog, VHDL) для синтезируемого HDL описания
 - Выбрать язык для модели на HDL
- Задать папку для описаний системы (**Output Directory**)
- Для запуска процедуры – используется кнопка **Generate**



Альтернативный способ запуска процедуры:

Меню *Generate* → *Generate HDL*

План

- О приложении PD
- Пользовательский интерфейс приложения PD
- **Файлы, создаваемые PD, при генерации системы**
- Место PD в процедуре проектирования пакета QP
- Лабораторная 1

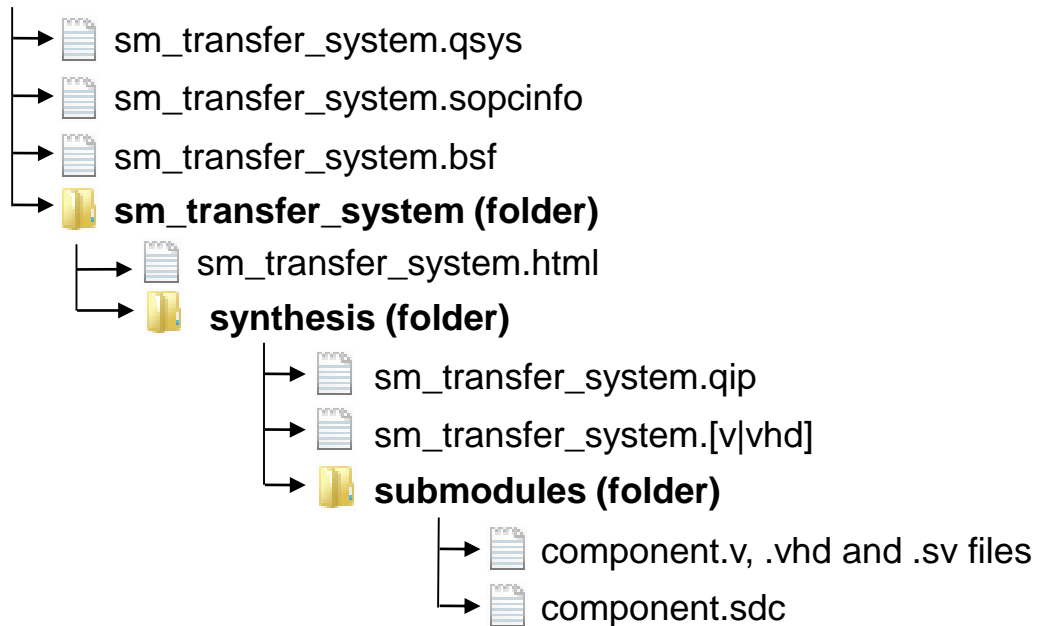
Структура папок, создаваемых при генерации

Структура одинакова для всех микросхем, выпущенных ранее Intel® Arria® 10



project (folder)

Система - sm_transfer_system



Файл .qsys

- Файл .qsys полностью описывает систему, созданную в Platform Designer (компоненты, подключения, параметры...)
- В рамках одного проекта Quartus (в одной рабочей папке проекта) может быть создано несколько систем.
- В папке где находится файл .qsys создается папка .qsys_edit
 - Содержит настройки системы и раскладку PD

Файлы создаваемые при генерации системы

- `<system_name>.sopcinfo`
 - XML файл, описывающий систему, созданную в Platform Designer system, используется при разработке ПО
- `<system_name>.bsf`
 - Файл с изображением символа системы. Используется при схемном вводе файла верхнего уровня в пакете Quartus (можно отключить создание)
- `<system_name>/<system_name>.html`
 - Отчет о результатах процедуры генерации HDL описания системы

Файлы создаваемые при генерации системы

- Файлы для синтеза
 - Папка - <system_name>/**synthesis**
 - <system_name>.**qip**
 - Скрипт, содержащий ссылки на файлы, необходимые для синтеза системы
 - <system_name>.[v|vhd]
 - Файл верхнего уровня в описании системы – связывает все компоненты
 - Папка <system_name>/synthesis/**submodules** - содержит HDL описания модулей системы, сгенерированные PD на основе исходных кодов компонентов IP
 - Описания могут быть представлены на языках Verilog, SystemVerilog или VHDL
 - Папка может содержать файл(ы) .sdc – файлы с требованиями к временным параметрам
- Файлы для моделирования
 - Папка - <system_name>/**simulation**

План

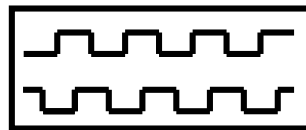
- О приложении PD
- Пользовательский интерфейс приложения PD
- Файлы, создаваемые PD, при генерации системы
- Место PD в процедуре проектирования пакета QP
- Лабораторная 1

Процедура проектирования

Design specification



Platform Designer



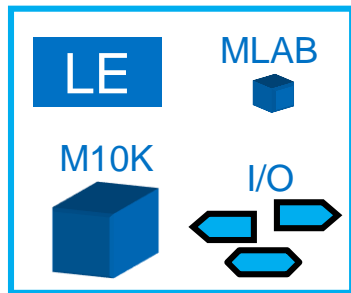
RTL simulation

- Functional simulation
- Verify logic model & data flow (no timing delays)



Synthesis

- Translate design into device specific primitives
- Optimization to meet required area & performance constraints
- Intel® Quartus® Prime software or other supported synthesis tools

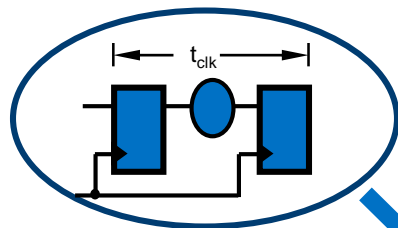


Intel® Quartus® Prime
Design Software

Place & Route

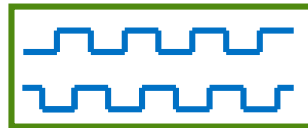
- Map primitives to specific locations inside
- Target technology with reference to area & performance constraints
- Specify routing resources to be used

FPGA Hardware Design Flow (cont.)



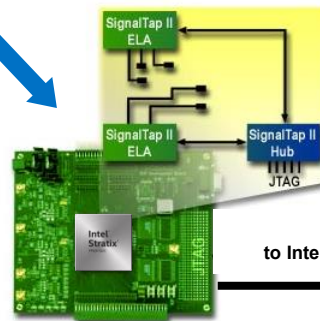
Timing analysis

- Verify performance specifications were met
- Timing analyzer static timing analysis



Gate-level simulation

- Functional timing simulation
- Verify design will work in target technology
- Get signal toggle info for power analysis
- *Combo of RTL simulation and timing analysis usually sufficient for verification*



Test FPGA on PC Board

- Program & test device on board
- Use tools like Signal Tap logic analyzer & System Console for debugging

Download cable
to Intel® Quartus® Prime software

Процедура проектирования с использованием PD

1. Создать проект в пакете QP
2. Запустить PD и создать систему
3. В PD запустить процедуру генерации HDL описаний системы (и сопутствующих файлов)
 - Папка по умолчанию `<project folder>/<system name>`
4. Подключить к проекту файлы: **.qsys** или **.qip**
 - **.qsys**: созданная система будет автоматически регенерироваться каждый раз, когда компилируется (синтезируется) проект в пакете Quartus.
 - **.qip**: к проекту добавляются сгенерированные HDL файлы с описанием системы, регенерация системы, если система была изменена, запускается в PD
5. В QP создать файл верхнего уровня проекта системы (это может быть файл-обертка, только переименовывающий выводы системы, созданной в PD)
6. В пакете ModelSim осуществить моделирование созданного файла верхнего уровня
 - *Потребуется создание теста*

Процедура проектирования с использованием PD

6. Осуществить реализацию и отладку созданного файла верхнего уровня проекта системы на плате
 - Используется InSystemSource &Probe; SignalTapII
 - Потребуется создание файла с описанием системы для отладки
7. Задать требования к проекту системы:
 - к временным параметрам - timing constraints (**.sdc**),
 - подключению выводов микросхемы к сигналам модуля верхнего уровня (**.qsf**)
 - Многие IP имеют свой набор требований (их файл **.sdc** указан в **.qip** файле)
 - Необходимо создать файл **.sdc** для модуля верхнего уровня, как минимум для:
тактовых входов (`create_clock`, `create_generated_clock`) и тактовых сигналов,
порожденных в PLL (`derive_pll_clocks`)
8. Осуществить полную компиляцию проекта

План

- О приложении PD
- Пользовательский интерфейс приложения PD
- Файлы, создаваемые PD, при генерации системы
- Место PD в процедуре проектирования пакета QP
- Лабораторная 1