

Проектирование в QSys с применением процессора Nios II

Программа курса

- Описание программного процессорного ядра Nios® II
- Использование Qsys для создания сложных систем
- Разработка и отладка программного обеспечения для процессора Nios II
- Интерфейсы систем на кристалле
- Разработка пользовательских периферийных модулей
- Разработка пользовательских инструкций
- Моделирование системы в среде ModelSim®
- Запуск и отладка проекта
- Программирование отладочной платы

Продукция Altera

ALTERA®



Аппаратные и программные процессорные ядра

Nios® II
ARM®

Средства проектирования



Комплекты разработчика

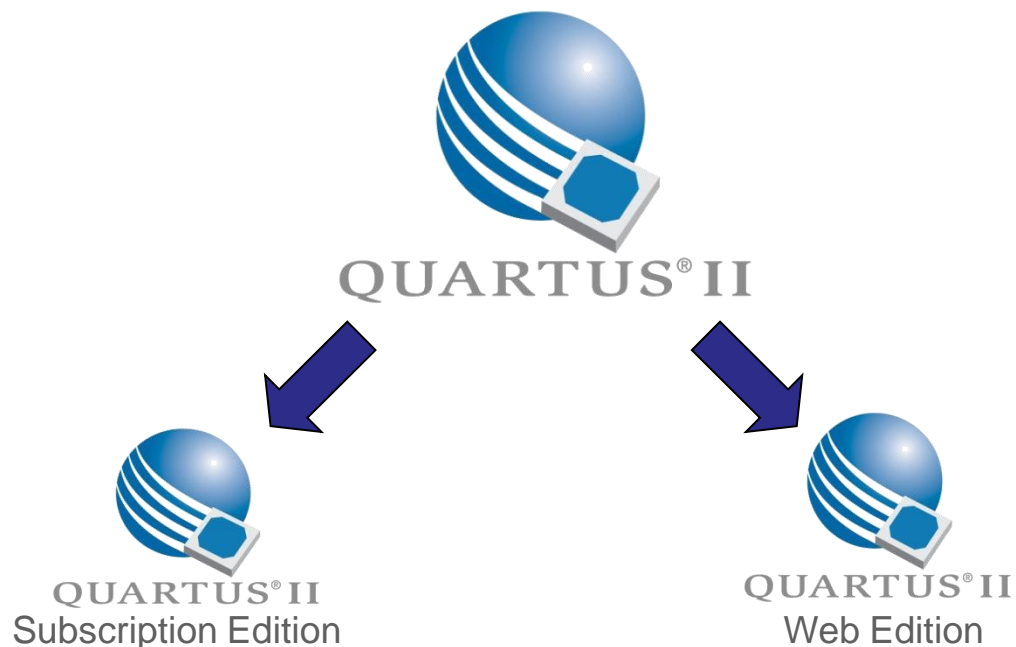


IP ядра

- Industrial
- Computing
- Enterprise



Две конфигурации Quartus II



Поддержка устройств

Все

Ограничена

Возможности

100%

95%

Распространение

Интернет и DVD

Интернет и DVD

Стоимость

Платная

Бесплатно

[Подробное сравнение возможностей приведено
на сайте Altera](#)

Требования к установке

- Последняя версия Quartus II
- Профиль пользователя не должен содержать русских букв
- Не использовать пробелы и русские буквы в имени пути проекта
- Не использовать пробелы и русские буквы в пути установки Quartus и Nios II
- Лицензия на Nios II

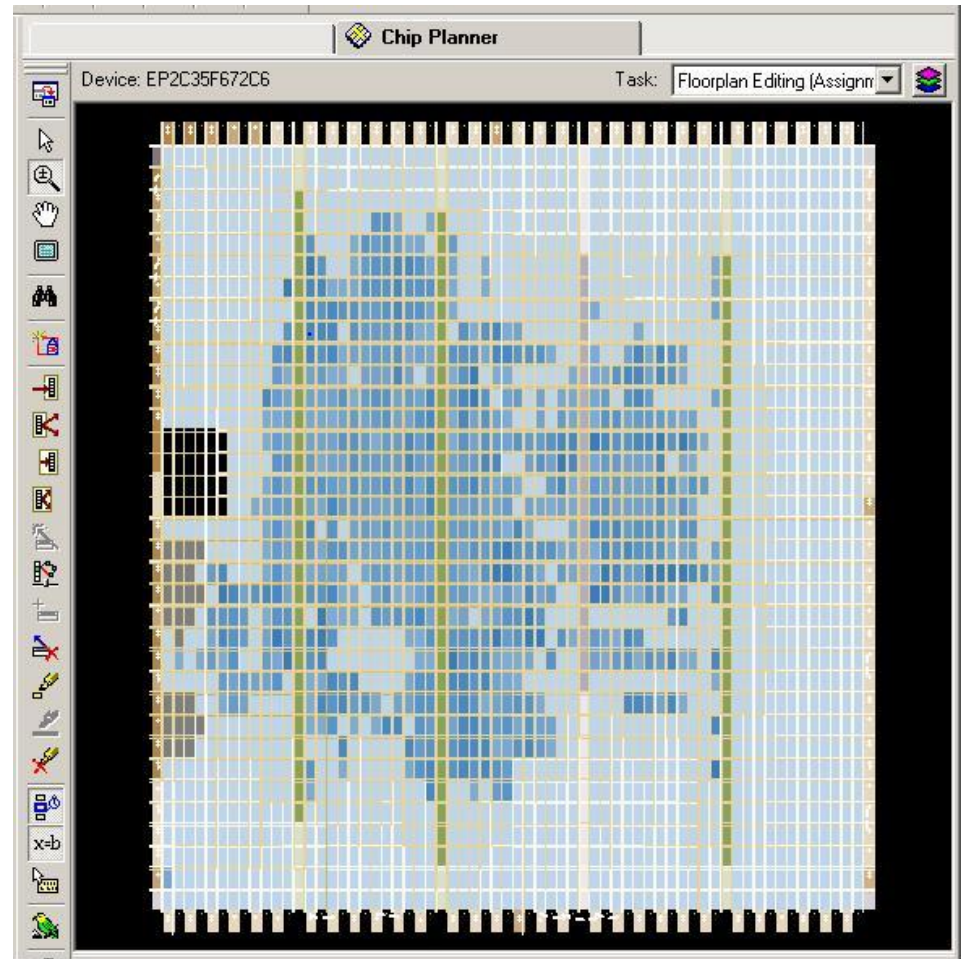
или

- Загрузочный кабель, соединенный с ПК для работы в режиме OpenCore Plus

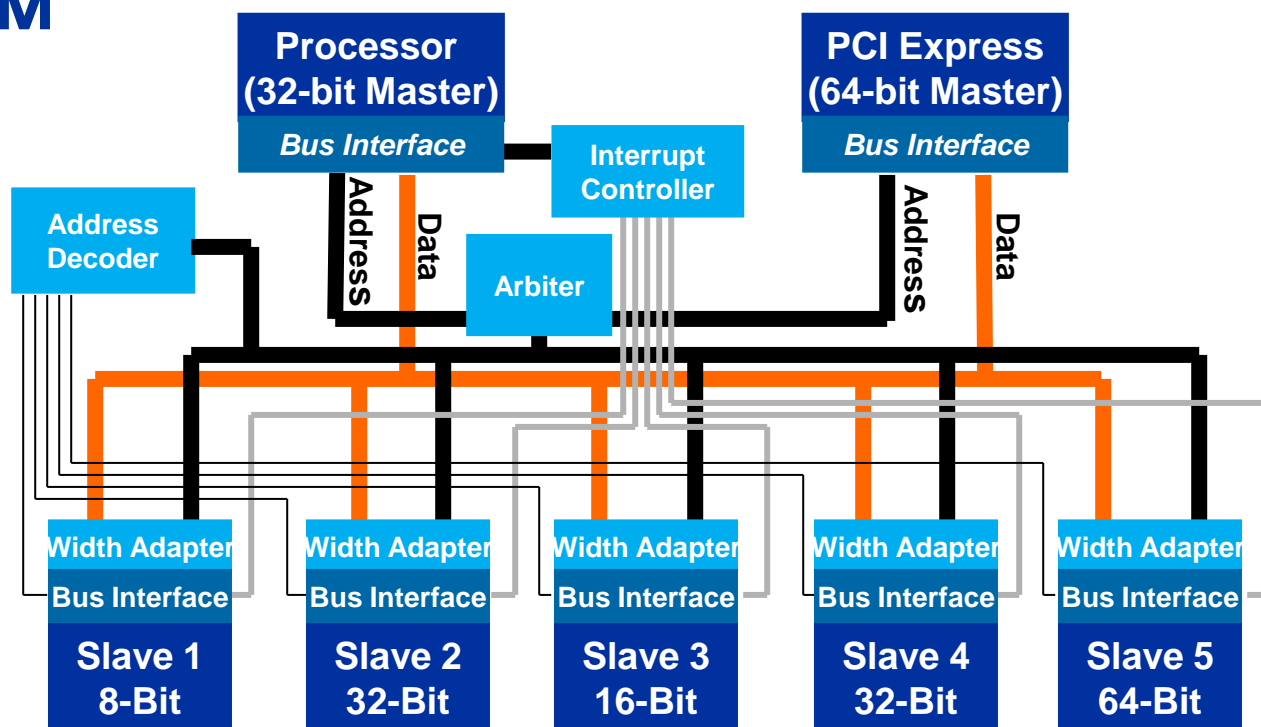
Возможности QSys

Система на кристалле в FPGA

- Полностью интегрированный маршрут проектирования в САПР Quartus II с использованием QSys

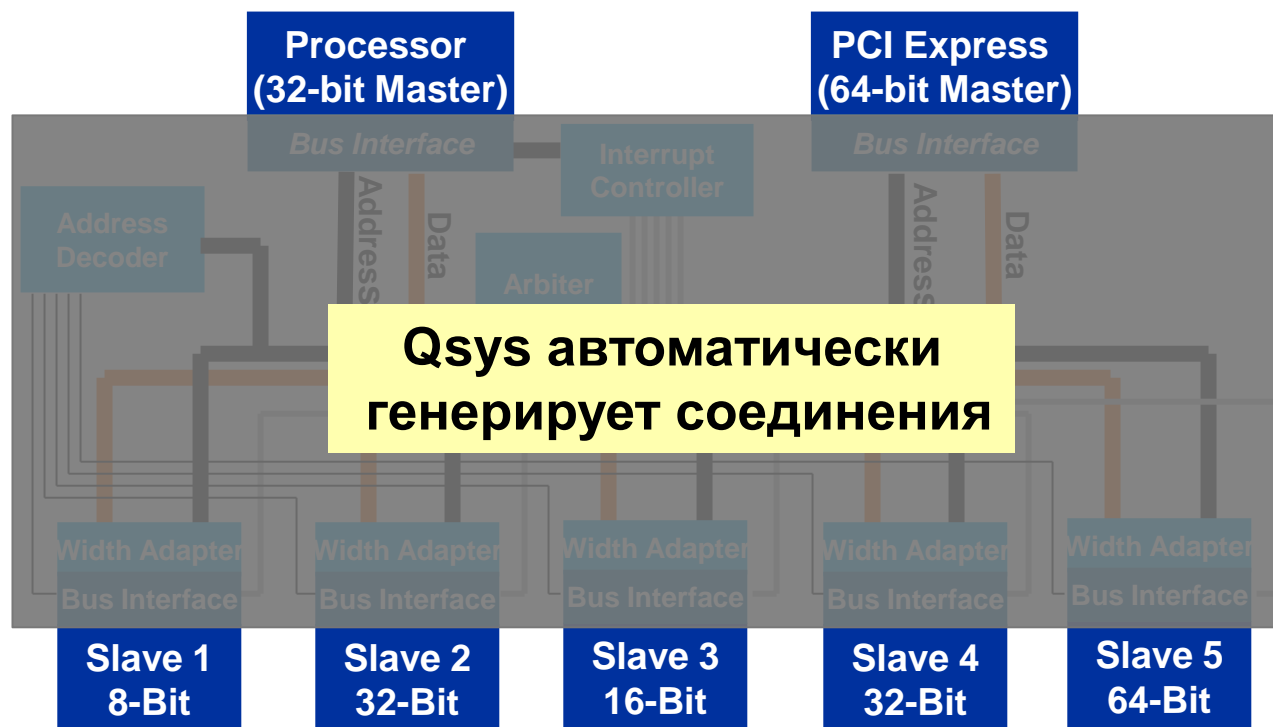


Традиционный подход к проектированию систем



- Компоненты могут использовать разные интерфейсы
- Требуется значительный объем усилий по проектированию соединений и обмена между модулями
- Интеграция модулей не автоматизирована и неудобна (в том числе стандартных)

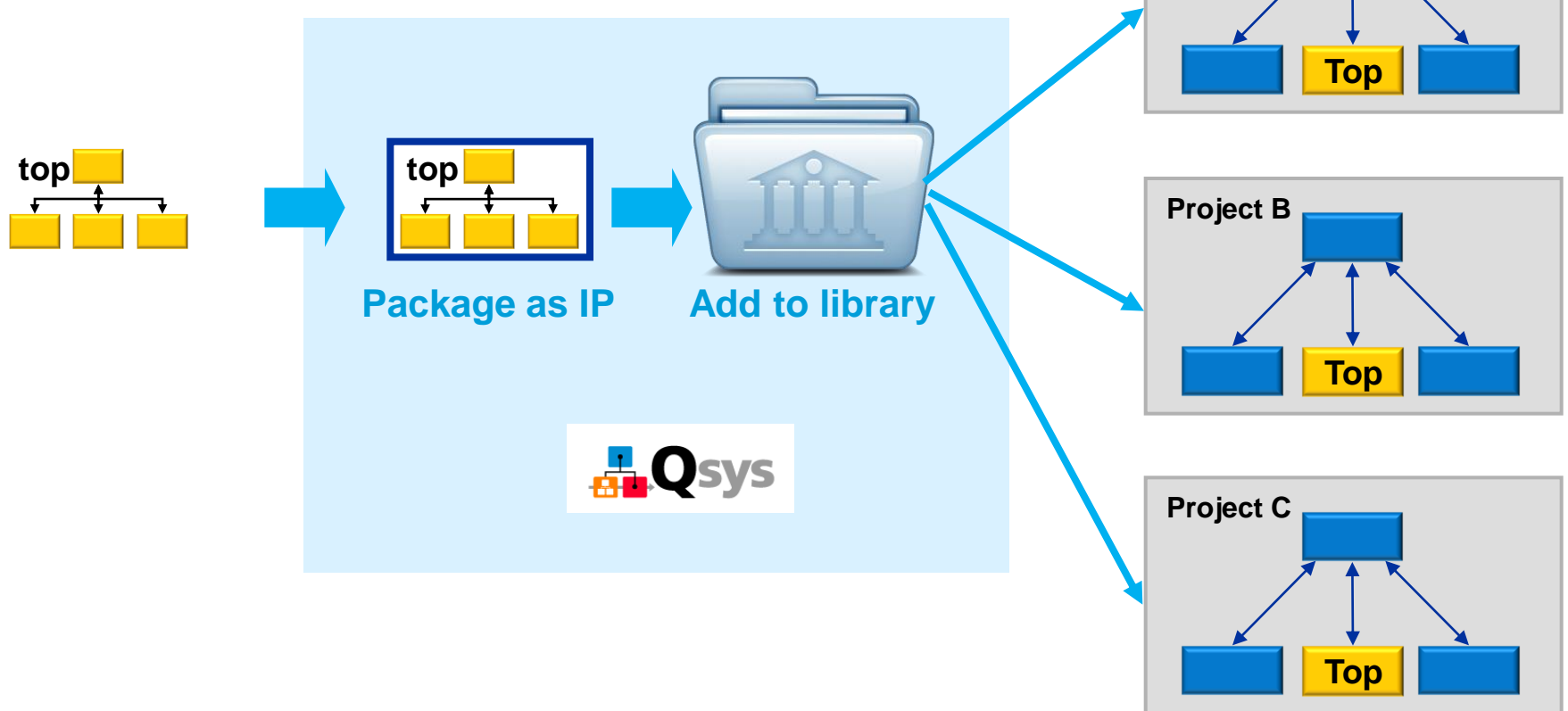
Автоматическая генерация соединений



- Уменьшает вероятность ошибок в процессе интеграции
- Сокращает время разработки
- Необходимо проектировать только нестандартные модули

Повторное использование проектов

- QSys обеспечивает возможность повторного использования IP и иерархического проектирования



Применение QSys

- Может использоваться практически в любом проекте на FPGA
- Две категории связей в проекте
 - Управление
 - С отображением в общую память
 - Чтение регистров статуса и запись регистров управления
 - Передача данных
 - Поточковая передача
 - Коммутация данных, объединение, мосты и FIFO
- Применения включают обработку видео и изображений, высокоскоростные интерфейсы, встроенные системы, интерфейс с внешней памятью

Управление

Что, если процессорное ядро не нужно?

- Системы Qsys не требуют процессора
- Другие компоненты могут формировать запросы на передачу и осуществлять чтение/запись регистров
 - Конечный автомат как главное устройство на общей шине
 - Поточковая обработка
- Внутренние процессоры (Nios[®] II, ARM[®] Cortex[™]-A9) могут быть интегрированы в Qsys

Преимущества Qsys

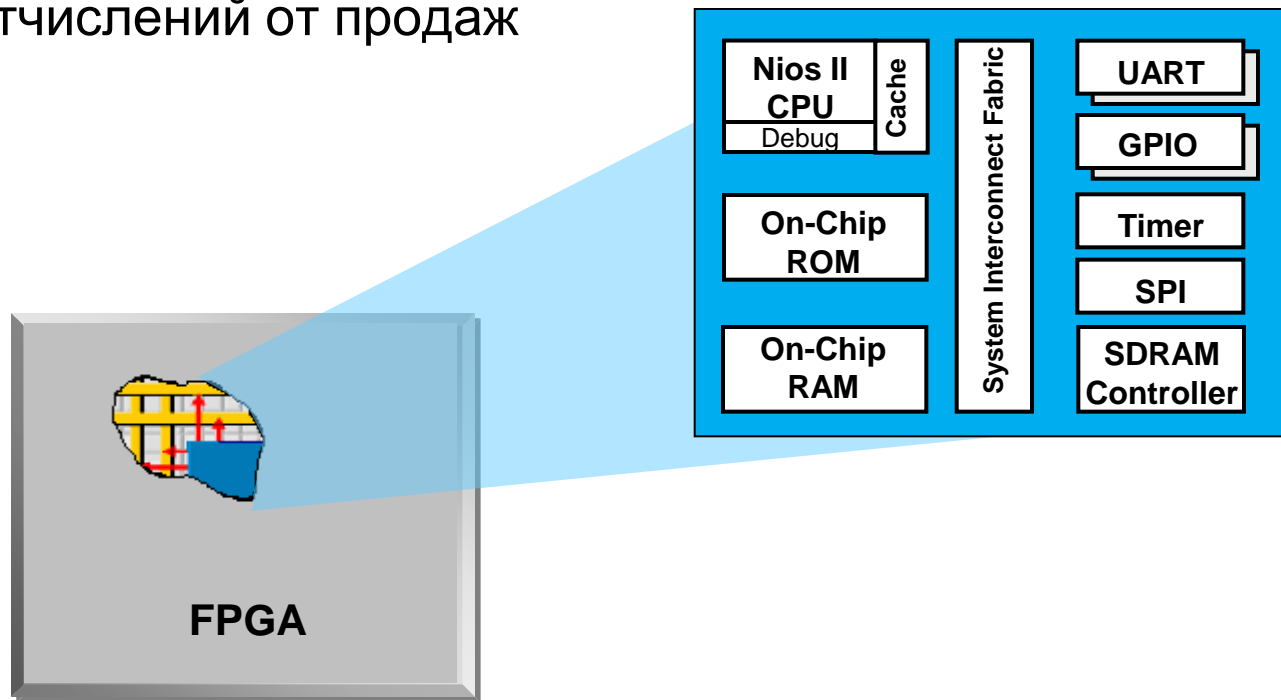
- Упрощает разработку сложных систем
 - Автоматическая генерация соединений
- Повышает уровень абстракции
 - Высокоуровневый ввод и визуализация
- Стандартная платформа
 - Для интеграции готовых IP ядер и модулей пользователя
 - Для верификации
- Возможность повторного использования модулей
- Простота масштабирования

Процессор Nios II

Введение

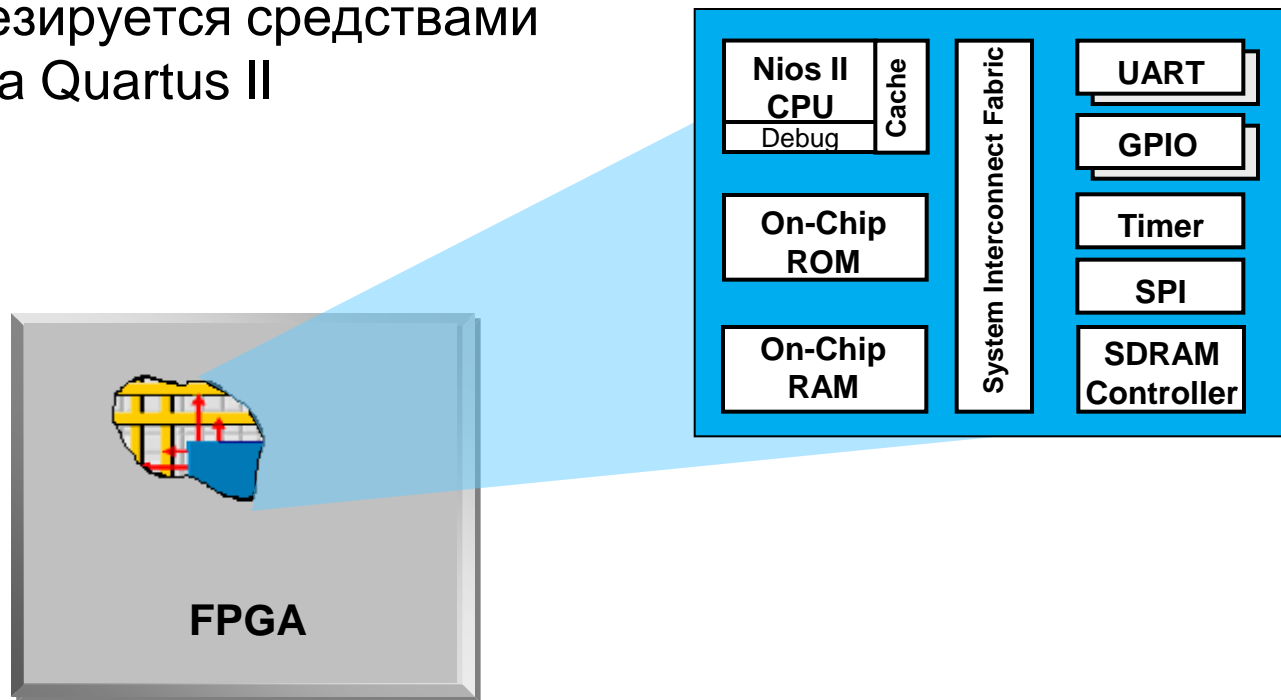
Что такое процессор Nios II ?

- Второе поколение программных 32-х разрядных RISC микропроцессоров
 - Собственная разработка Altera
 - Гарвардская архитектура
 - Без отчислений от продаж



Что такое процессор Nios II ?

- Второе поколение программных 32-х разрядных RISC микропроцессоров
 - Процессор Nios II и вся периферия реализованы на HDL
 - Может быть реализован в любых FPGA фирмы Altera
 - Синтезируется средствами пакета Quartus II



Архитектура процессора Nios II

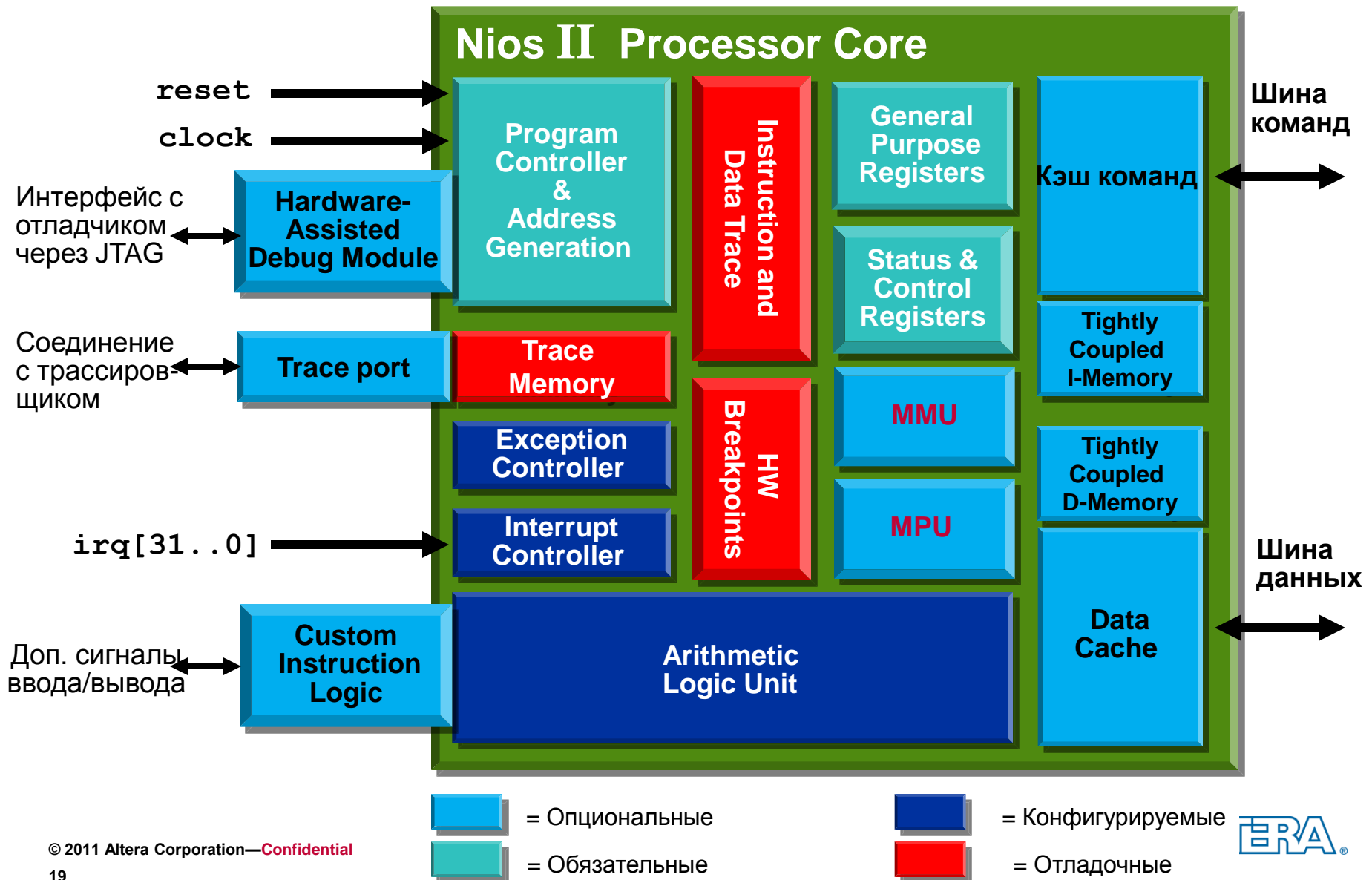
■ Классический конвейеризованный RISC процессор

- 32 регистра общего назначения
- 3 формата команд
- 32-х битные команды
- 32-х битные шины данных
- Раздельный кэш команд и данных (настраиваются)
- Тесно связанная память (TCM)
- Предсказание ветвлений
- 32 источника прерываний с приоритетом
- Аппаратная реализация умножения и сдвигов
- Модуль управления памятью (MMU)
- Модуль защиты памяти (MPU)
- Пользовательские инструкции
- Модуль отладки с поддержкой JTAG

Архитектура процессора Nios II

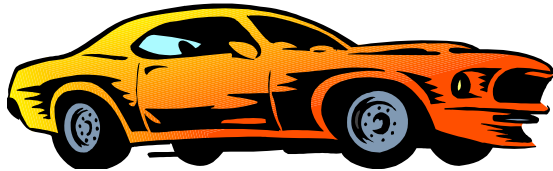
- **Адресация до 2Гбайт по шине данных без MMU**
 - Старший бит адреса зарезервирован как флаг обхода кэша
 - На практике максимальный объем модуля памяти составляет 1Гбайт, так как требуется выделить область в карте памяти для периферийных модулей
- **Адресация до 4Гбайт по шине данных с MMU**
- **Адресация до 256 Мбайт по шине инструкций**
 - Следует учитывать при проектировании подсистемы памяти

Блок-схема процессора Nios II



Версии процессора Nios II

■ Три версии процессора Nios II



- FAST: Оптимизирован по быстродействию



- STANDARD: Сбалансированная реализация



- ECONOMY: Оптимизирован по объему

■ Программное обеспечение

- Совместимо на уровне машинного кода
 - Не требуется изменения ПО при смене версии

Сравнение версий Nios II

	Nios II /f Fast	Nios II /s Standard	Nios II /e Economy
Конвейер	6 стадий	5 стадий	Нет
Аппаратный умножитель и сдвигатель	1 такт	3 такта	Эмулируется в ПО
Предсказание ветвлений	Динамическое	Статическое	Нет
Кэш команд	Конфигурируется	Конфигурируется	Нет
Кэш данных	Конфигурируется	Нет	Нет
Объем (ЛЭ)	1800 без MMU 3200 с MMU	1200	600
Пользовательские инструкции	До 256		

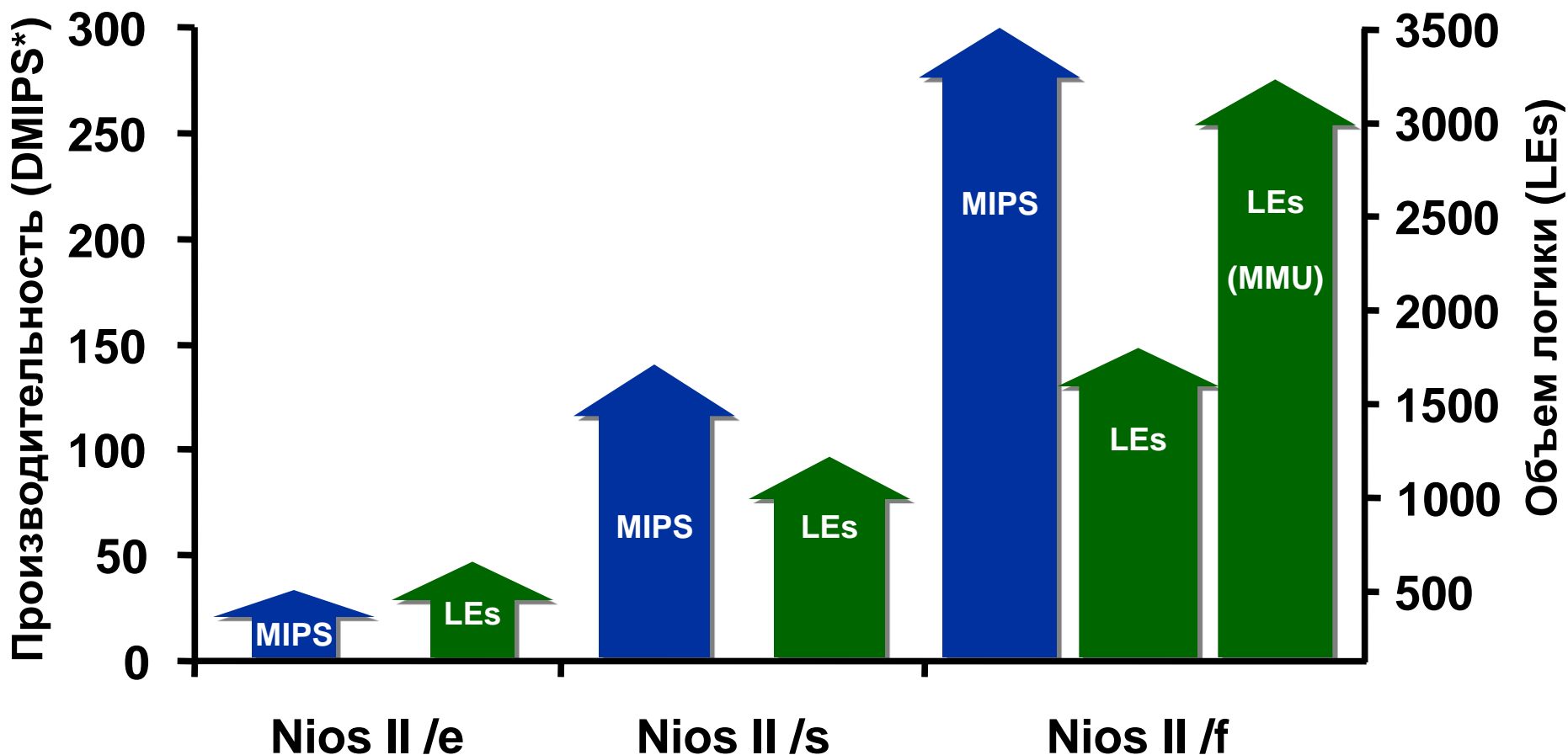
Аппаратный умножитель

- Nios II, версия Economy – нет аппаратного умножителя
 - Использует библиотеку GNUPro Math для реализации умножения
- Nios II, версия Standard – аппаратный умножитель
 - $32 \times 32 \rightarrow 32$ за 3 такта, если в семействе есть аппаратные умножители*
- Nios II Processor, Fast - Full Hardware Multiplier
 - $32 \times 32 \rightarrow 32$ за 1 такт, если в семействе есть аппаратные умножители*

Версия	Тактов ($32 \times 32 \rightarrow 32$)
Economy	250
Standard	3
Fast	1

*) Также может реализовываться аппаратно на логических элементах, умножение $32 \times 32 \rightarrow 32$ занимает 11 тактов

Производительность и объем разных версий процессора Nios II



* Dhrystone 2.1 Benchmark

Данные приведены для Stratix III

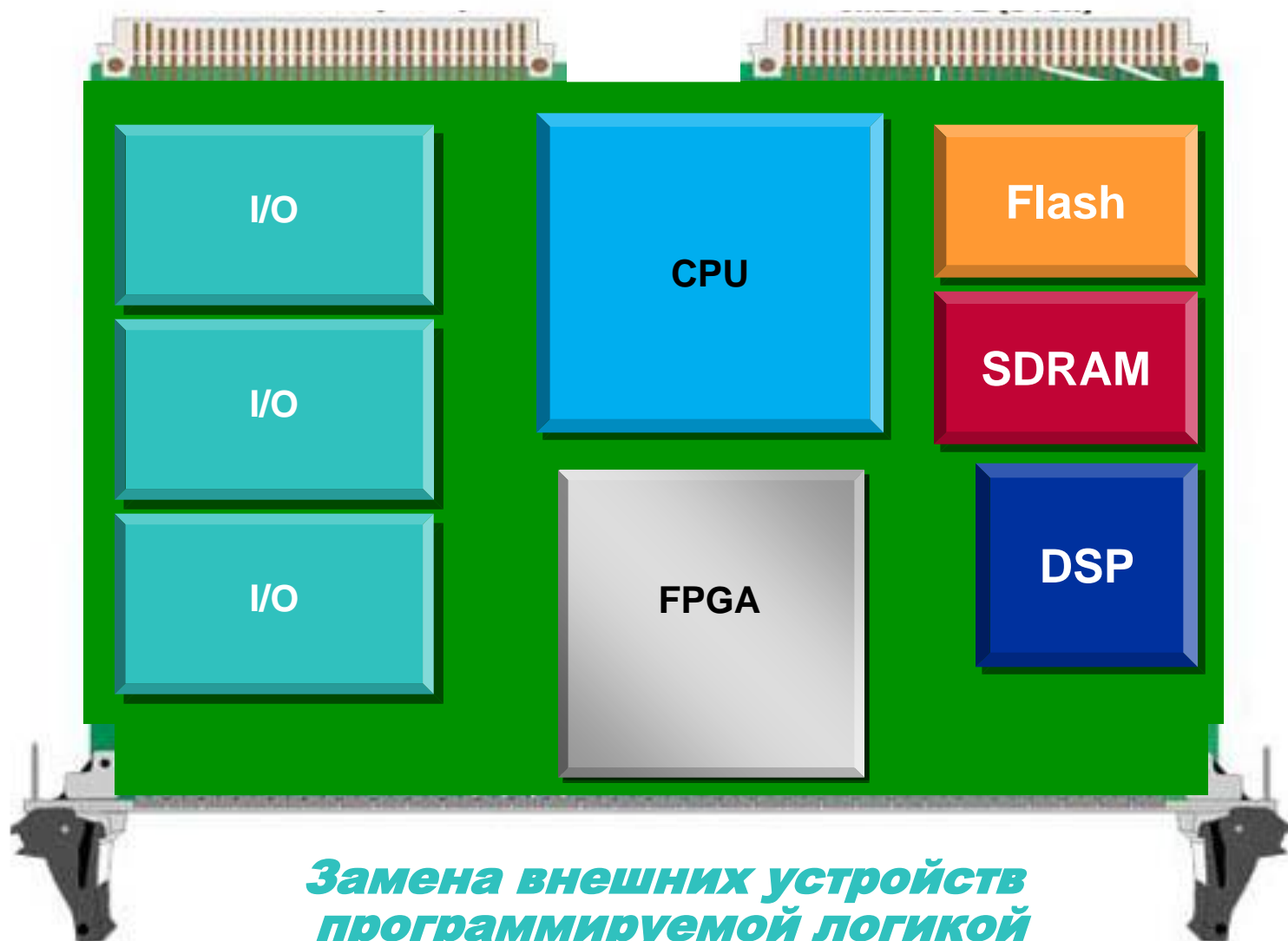
Производительность Nios II (МГц/DMIPS)

Семейство	Fast		Standard		Economy	
Stratix V	310	350	300	192	340	51
Arria V	160	180	180	115	190	28
Cyclone V	160	180	140	89	200	30
Stratix IV	240	271	230	147	290	43
Cyclone IV GX	160	181	130	83	170	26
Arria II GX	170	192	170	108	300	45
Cyclone III	175	195	145	90	215	30

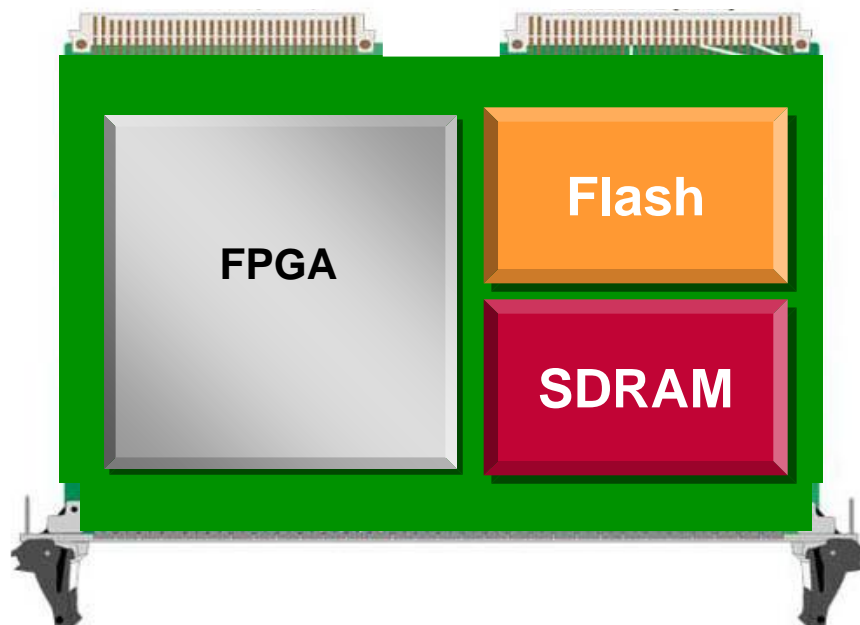
Лицензирование

- Процессор Nios II поставляется как зашифрованное IP ядро
 - Лицензия включается в файл лицензии пакета Quartus II
 - Ядро включает средство проверки наличия процессора Nios II в проекте пользователя
 - При отсутствии лицензии работает в режиме OpenCore Plus
 - Система работает неограниченно долго в режиме отладки, когда есть соединение со средствами проектирования на ПК
 - Система перестает работать примерно через один час если нет связи с ПК
 - При наличии активной подписки
 - Подписчики и покупатели средств проектирования получают лицензию а www.altera.com
 - Проект, откомпилированный с такой лицензией, работает без ограничений
- Лицензия с открытым исходным кодом
 - Доступна по запросу, вопрос о предоставлении решается фирмой Altera
 - Требуется при переносе проектов на устройства других фирм и полузаказные СБИС

Уменьшение цены, сложности и потребления



System On A Programmable Chip (SOPC)

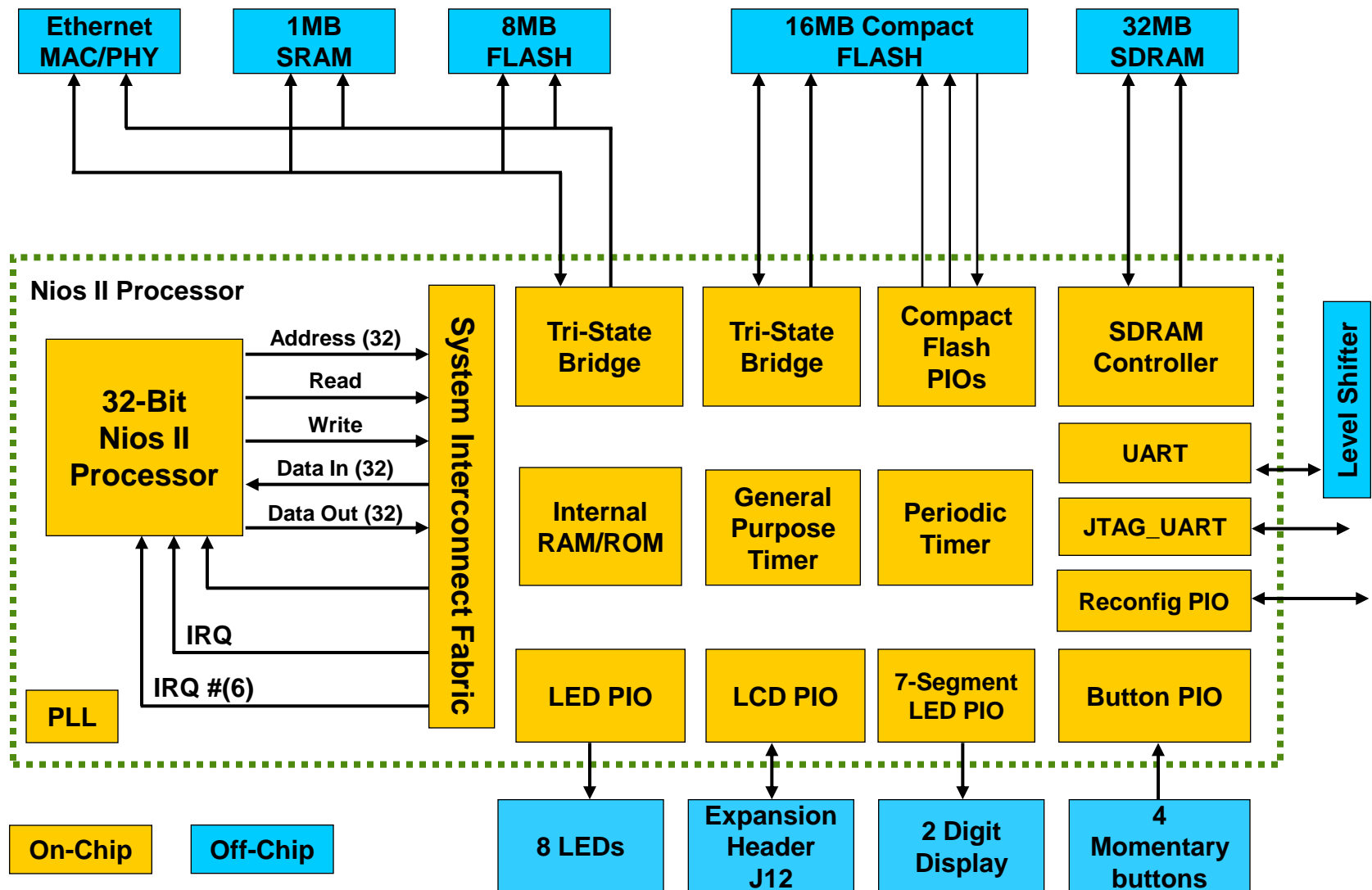


***Наличие процессорного ядра требуется
для решения задач управления при системной
интеграции***

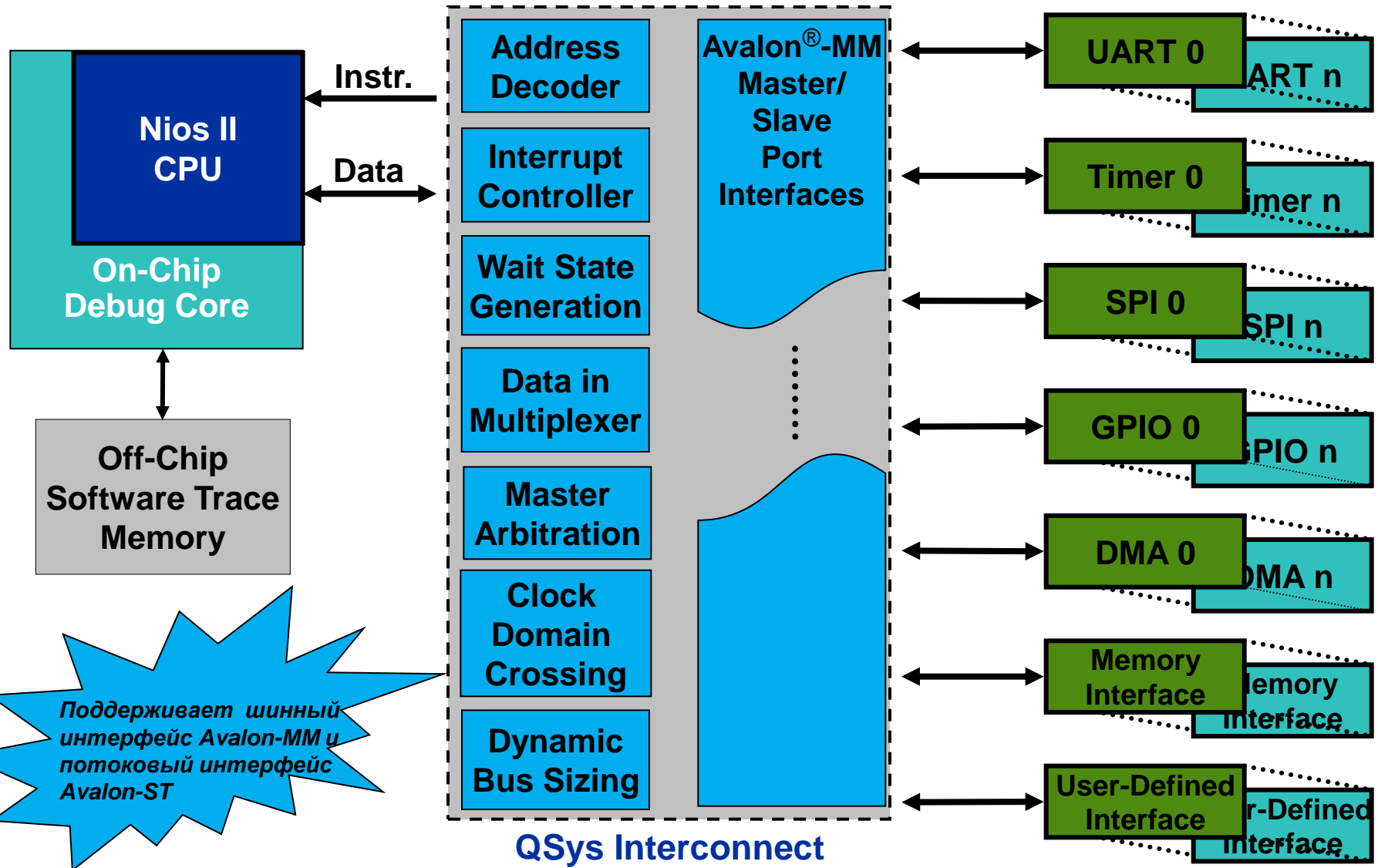
Комплекты разработчика



Блок-схема базового проекта



Структура системы на процессоре Nios II



Другие комплекты разработчика

- http://www.altera.com/products/devkits/kit-dev_platforms.jsp
- Комплекты разработчика от Altera, Terasic, EBV Elektronik и других производителей
- Доступные и специализированные комплекты
- Схема и реализация могут использоваться как reference design при проектировании

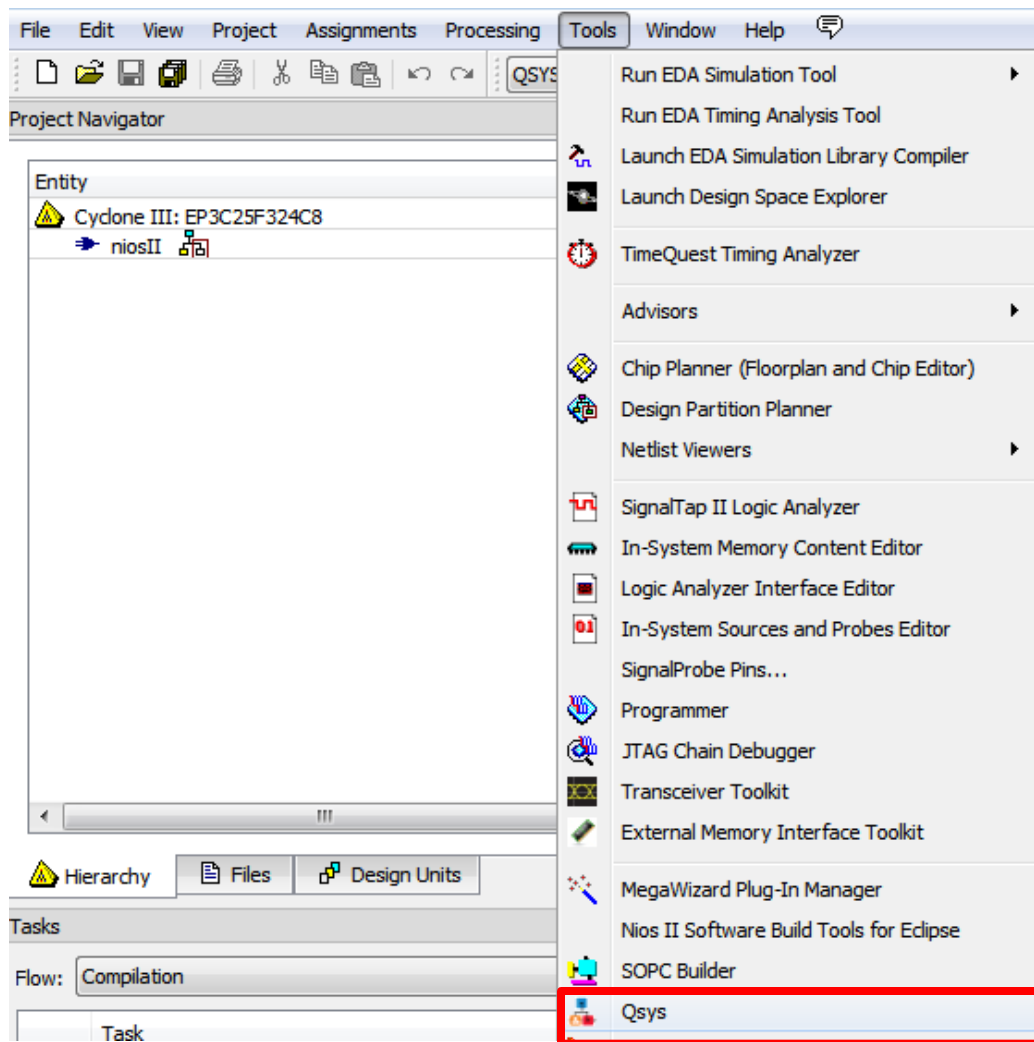
Базовые проекты для комплектов разработчика

- Можно найти проекты
 - На Altera.com
<http://www.altera.com/products/devkits/kit-index.html>
 - На Altera Wiki
<http://www.alterawiki.com/wiki/Special:Categories>
- Могут использоваться как есть в качестве аппаратной платформы (прошивки ПЛИС) или как отправная точка для создания системы на кристалле

Создание системы в QSys

Запуск QSys

- Создайте проект в Quartus II
- Запустите Qsys из меню **Tools**
- Откройте или создайте новую систему в QSys



Главная страница QSys

The screenshot shows the QSys IDE main page with the following components and annotations:

- Library**: A sidebar on the left containing a **Project** section with 'New Component...' and a **Library** section with various components like Bridges, Clock and Reset, DSP, etc. Below this is a **Hierarchy** section showing a tree of modules for 'niosii', including 'btn', 'clk', 'flash_bus', 'hex', 'led', 'ram_bus', 'reset', 'buttons', 'cpu', 'ext_flash_ctl', 'ext_ram_bus', 'ext_ram_ctl', 'green_led', 'high_res_timer', and 'clk'. An annotation **Библиотека компонент** points to the Library section, and **Иерархия модулей** points to the Hierarchy section.
- System Contents**: A central pane showing a system diagram with components like 'clk', 'reset', 'tcs', 'out', 'jtag_uart', 'seven_seg', 'green_led', 'buttons', and 'sys_clk_timer'. An annotation **Система** points to this pane.
- Address Map**: A pane on the right showing a table of components and their properties. An annotation **Закладки с настройками (выбор в меню View)** points to the tabs at the top of this pane.
- Messages**: A pane at the bottom showing a list of messages, including 2 warnings and 3 info messages. An annotation **Сообщения** points to this pane.

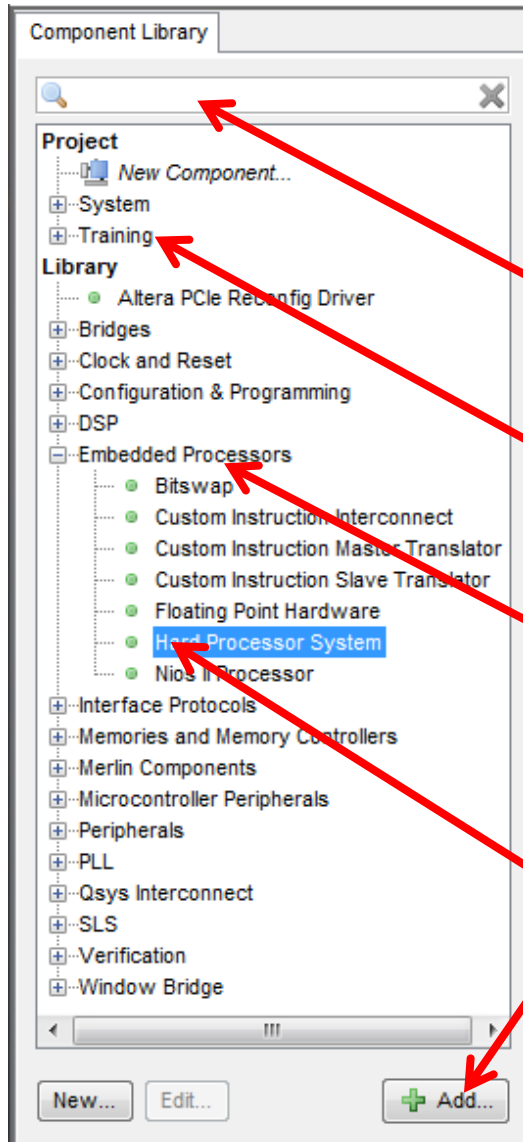
Use	Connections	Name	Description	Export	Clock
		clk	Clock Input	Double-click to export	clk
		reset	Reset Input	Double-click to export	[clk]
		tcs	Tristate Conduit Slave	Double-click to export	[clk]
		out	Conduit		
<input checked="" type="checkbox"/>		jtag_uart	JTAG UART		
		clk	Clock Input	Double-click to export	clk
		reset	Reset Input	Double-click to export	[clk]
<input checked="" type="checkbox"/>		seven_seg	PIO (Parallel I/O)	Double-click to export	clk
		avalon_jtag_slave	Avalon Memory Mapped Slave	Double-click to export	[clk]
		clk	Reset Input	Double-click to export	clk
		rese:	Avalon Memory Mapped Slave	Double-click to export	[clk]
		s1	Conduit	Double-click to export	clk
		external_connection	Conduit	Double-click to export	hex
<input checked="" type="checkbox"/>		green_led	PIO (Parallel I/O)	Double-click to export	clk
		clk	Clock Input	Double-click to export	clk
		reset	Reset Input	Double-click to export	[clk]
		s1	Avalon Memory Mapped Slave	Double-click to export	clk
		external_connection	Conduit	Double-click to export	clk
<input checked="" type="checkbox"/>		buttons	PIO (Parallel I/O)	Double-click to export	clk
		clk	Clock Input	Double-click to export	clk
		reset	Reset Input	Double-click to export	[clk]
		s1	Avalon Memory Mapped Slave	Double-click to export	clk
		external_connection	Conduit	Double-click to export	clk
<input checked="" type="checkbox"/>		sys_clk_timer	PIO (Parallel I/O)	Double-click to export	clk
		clk	Clock Input	Double-click to export	clk
		reset	Reset Input	Double-click to export	[clk]

■ Ядра Altera и партнеров Altera

- Процессоры
- Интерфейсы памяти
- Периферия
- Мосты
- Аппаратные ускорители

■ Пользовательские модули

Библиотека компонент



- Перечень доступных IP-ядер и подсистем QSys

Можно ввести часть названия для поиска в списке

Использование существующих систем (иерархическое проектирование)

Компоненты распределены по категориям

Добавление компонента осуществляется двойным щелчком или нажатием на кнопку Add

Закладка System Contents

- Отображает компоненты и подсистемы в составе системы
- Используется для добавления компонент, их связи и настройки

System Contents Address Map Project Settings

Use	Connections	Name	Description	Export	Clock
<input type="checkbox"/>		uas	Avalon Memory Mapped Slave	Double-click to export	[clk]
<input type="checkbox"/>		tcm	Tristate Conduit Master	Double-click to export	[clk]
<input checked="" type="checkbox"/>		ext_ram_bus	Tri-State Conduit Bridge		
<input type="checkbox"/>		clk	Clock Input	Double-click to export	clk
<input type="checkbox"/>		reset	Reset Input	Double-click to export	[clk]
<input type="checkbox"/>		tcs	Tristate Conduit Slave	Double-click to export	[clk]
<input type="checkbox"/>		out	Conduit	Double-click to export	ram_bus
<input checked="" type="checkbox"/>		ext_flash_ctl	Generic Tri-State Controller		
<input type="checkbox"/>		clk	Clock Input	Double-click to export	clk
<input type="checkbox"/>		reset	Reset Input	Double-click to export	[clk]
<input type="checkbox"/>		uas	Avalon Memory Mapped Slave	Double-click to export	[clk]
<input type="checkbox"/>		tcm	Tristate Conduit Master	Double-click to export	[clk]
<input checked="" type="checkbox"/>		ext_flash_bus	Tri-State Conduit Bridge		
<input type="checkbox"/>		clk	Clock Input	Double-click to export	clk
<input type="checkbox"/>		reset	Reset Input	Double-click to export	[clk]
<input type="checkbox"/>		tcs	Tristate Conduit Slave	Double-click to export	[clk]
<input type="checkbox"/>		out	Conduit	Double-click to export	flash_bus
<input checked="" type="checkbox"/>		jtag_uart	JTAG UART		
<input type="checkbox"/>		clk	Clock Input	Double-click to export	clk
<input type="checkbox"/>		reset	Reset Input	Double-click to export	[clk]
<input type="checkbox"/>		avalon_jtag_slave	Avalon Memory Mapped Slave	Double-click to export	[clk]
<input checked="" type="checkbox"/>		seven_seg	PIO (Parallel I/O)		
<input type="checkbox"/>		input	Input	Double-click to export	clk
<input type="checkbox"/>		reset	Reset Input	Double-click to export	[clk]
<input type="checkbox"/>		avalon_slave	Avalon Memory Mapped Slave	Double-click to export	[clk]
<input type="checkbox"/>		conduit	Conduit	Double-click to export	hex

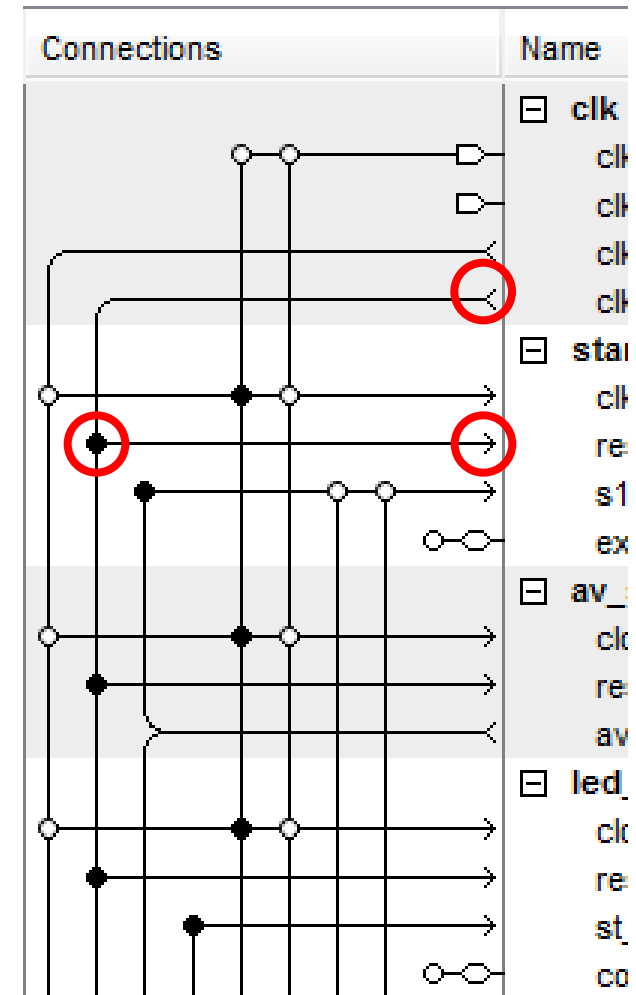
Компоненты

**Включить/временно
исключить из
состава системы**

ext_flash_ctl.uas
Avalon Memory Mapped Slave [avalon_slave 13.1]

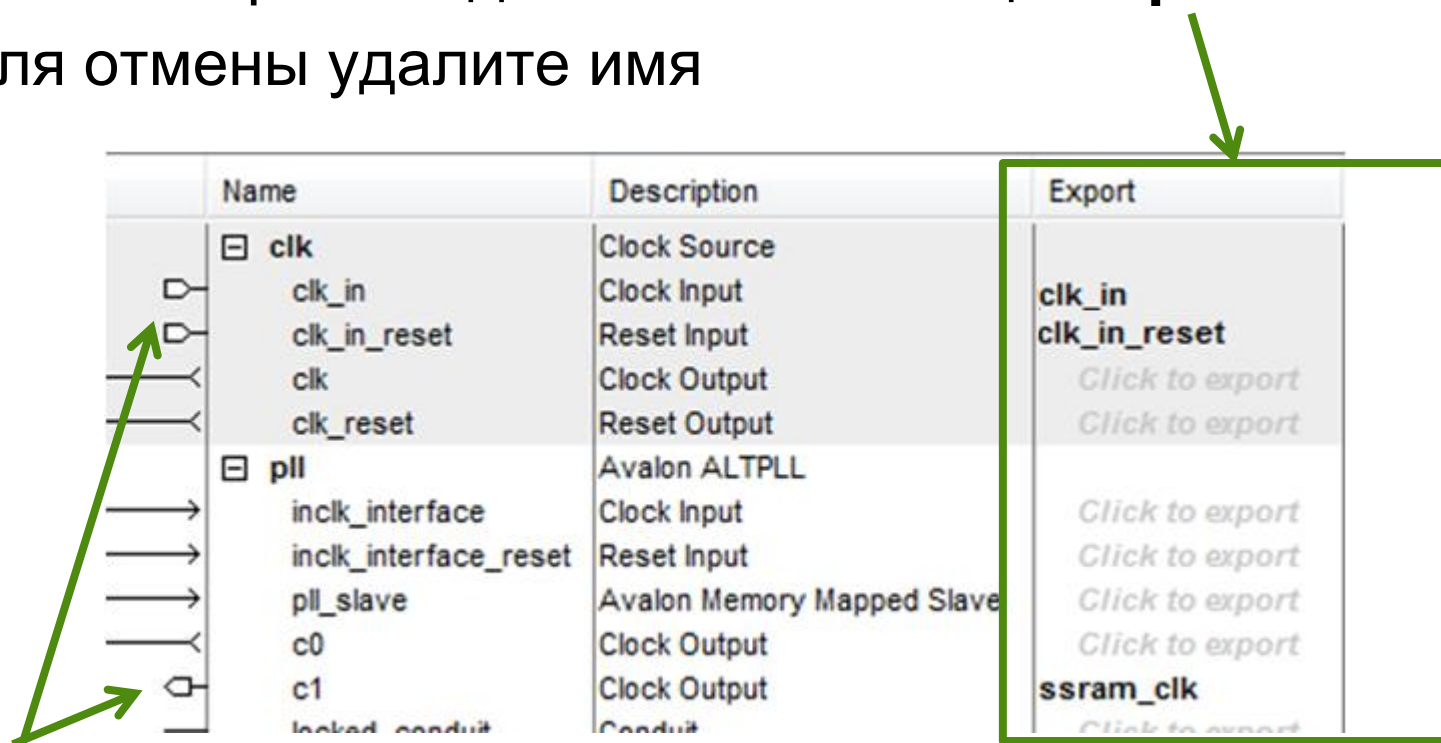
Столбец соединений Connections

- Столбец соединений используется для связывания компонент по различным интерфейсам
 - Тактовые импульсы и сброс
 - Интерфейсы передачи данных
 - ...
- Точка означает наличие соединения
- Направление соединения отражается стрелкой
 - От главного к подчиненному устройству
 - От источника к приемнику
- Имеется возможно скрыть столбец
 - Вызовите контекстное меню на заголовке "Connections"



Экспорт интерфейсов

- Экспорт интерфейсов из системы на кристалле
 - Может быть экспортирован любой интерфейс
- Для экспорта введите имя в столбце **Export**
- Для отмены удалите имя



	Name	Description	Export
	clk	Clock Source	
□	clk_in	Clock Input	clk_in
□	clk_in_reset	Reset Input	clk_in_reset
	clk	Clock Output	<i>Click to export</i>
	clk_reset	Reset Output	<i>Click to export</i>
	pll	Avalon ALTPLL	
→	inclk_interface	Clock Input	<i>Click to export</i>
→	inclk_interface_reset	Reset Input	<i>Click to export</i>
→	pll_slave	Avalon Memory Mapped Slave	<i>Click to export</i>
□	c0	Clock Output	<i>Click to export</i>
□	c1	Clock Output	ssram_clk
	locked_conduit	Conduit	<i>Click to export</i>

Значок порта показывает экспортированные интерфейсы

Просмотр и редактирование карты памяти

apped Slave	<i>Click to export</i>	clk		
apped Slave	<i>Click to export</i>	[clk1]	0x02400000	0x02400fff
apped Slave	<i>Click to export</i>	[clk1]	0x02400000	0x02400fff
	<i>Click to export</i>	[clk2]		
	<i>Click to export</i>	clk		
	<i>Click to export</i>	[clk2]		
ral	<i>Click to export</i>	clk		
apped Slave	<i>Click to export</i>	[clk]		
apped Slave	<i>Click to export</i>	[clk]	0x01202070	0x01202077

Редактирование в System Contents

Закладка редактирования карты памяти Address Map

System Contents Address Map Project Settings			
	cpu.data_master	cpu.instruction_master	cpu.tightly_coupled_instruction_master_0
cpu.jtag_debug_module	0x0140_0800 - 0x0140_0fff	0x0140_0800 - 0x0140_0fff	
sysid.control_slave	0x0140_1070 - 0x0140_1077		
instruction_tcm.s1			0x0200_0000 - 0x0200_3fff
instruction_tcm.s2	0x0200_0000 - 0x0200_3fff		
ext_ram_ctl.uas	0x0120_0000 - 0x013f_ffff	0x0120_0000 - 0x013f_ffff	
ext_flash_ctl.uas	0x0080_0000 - 0x00ff_ffff	0x0080_0000 - 0x00ff_ffff	
jtag_uart.avalon_jtag_slave	0x0140_1078 - 0x0140_107f		
seven_seg.s1	0x0140_1040 - 0x0140_104f		
green_led.s1	0x0140_1050 - 0x0140_105f		
buttons.s1	0x0140_1060 - 0x0140_106f		
sys_clk_timer.s1	0x0140_1000 - 0x0140_101f		
high_res_timer.s1	0x0140_1020 - 0x0140_103f		

Интерфейсы Master перечислены в столбцах, подчиненные устройства — в строках

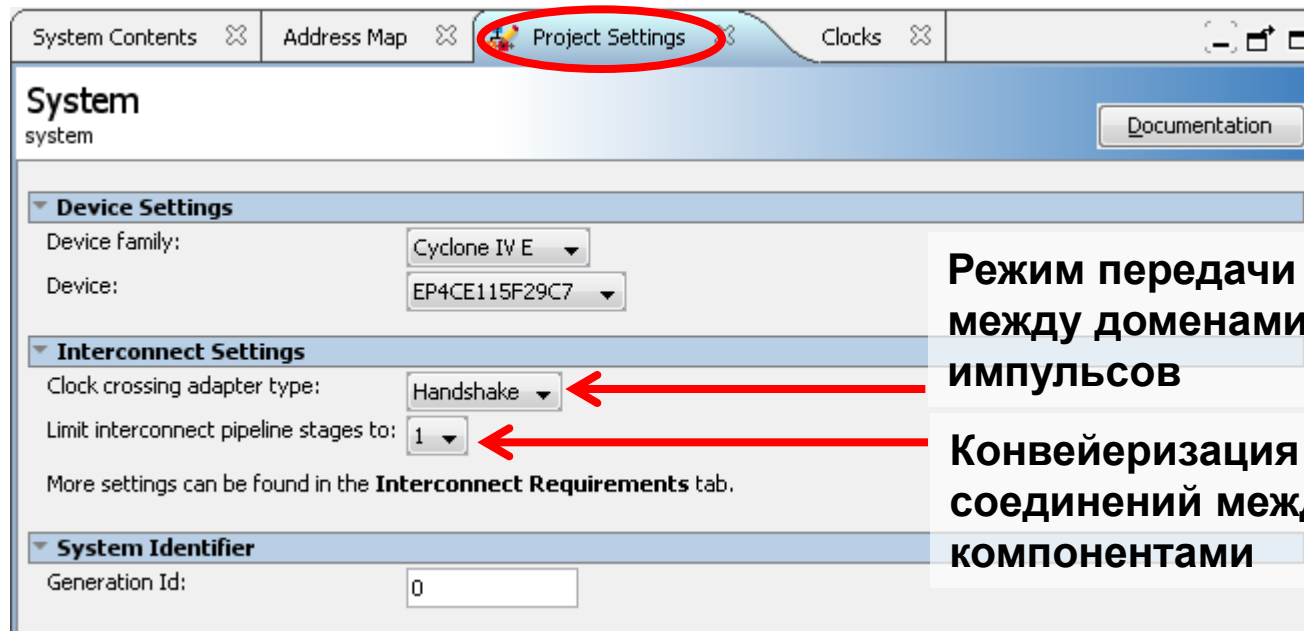
Закладка Clocks

- Вызов через View > Clocks
- Используется для:
 - Настройки тактовых импульсов
 - Добавления новых тактовых импульсов
 - Переименования тактовых импульсов
 - Указания частот (файл настроек TimeQuest в формате SDC создается пользователем)

Clock Settings		
Name	Source	MHz
clk	External	50.0
sys_clk	pll.c0	100.0
ssram_clk	pll.c1	100.0

Закладка Project Settings

- Выбор ПЛИС
- Управление генерацией межсоединений
 - Упрощает создание систем с разными ТИ
 - Настройка конвейеризации шины. Важно для быстродействия



Режим передачи данных
между доменами тактовых
импульсов

Конвейеризация
соединений между
компонентами

Закладка **System Inspector**

- Обзор системы в целом и информация о компонентах
 - Иерархия
 - Соединения верхнего уровня
 - Интерфейсы компонент
 - Соединения между компонентами
 - Информация о компонентах
- Редактирование настроек компонент

Закладки Hierarchy и Parameters

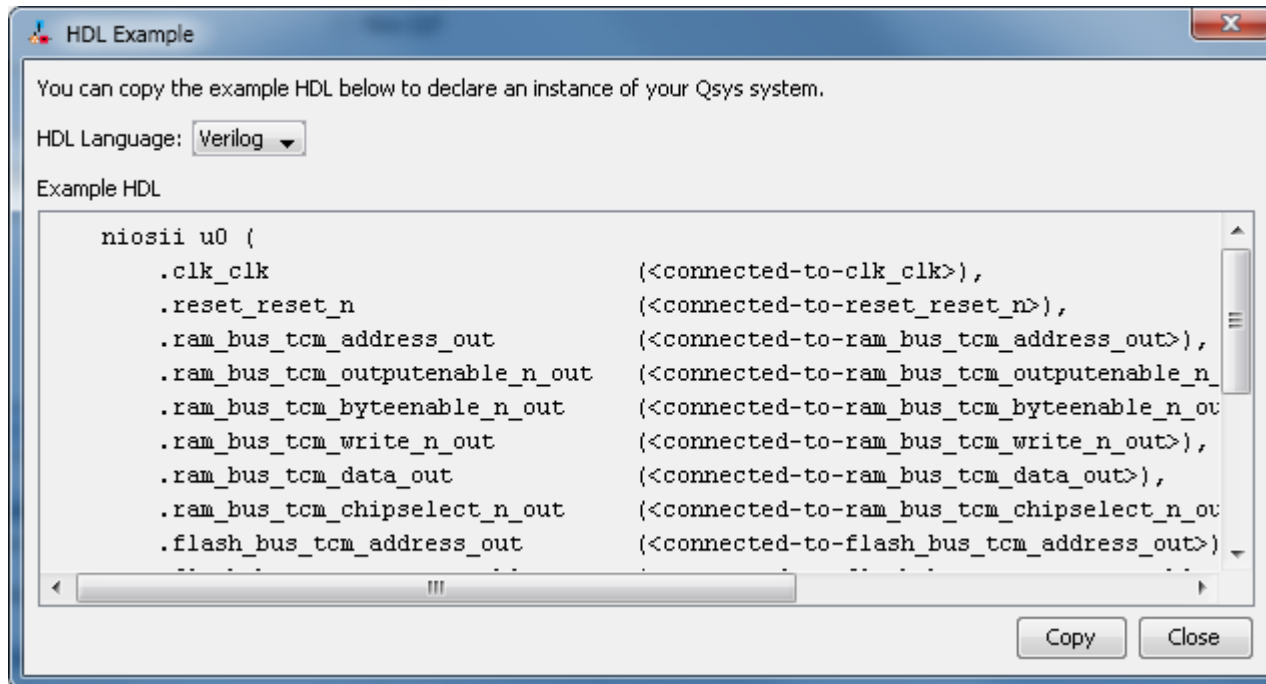
Выбор блока и просмотр его интерфейсов

Настройки блока

	Nios II/e	Nios II/s
Nios II Selector Guide	RISC 32-bit	RISC 32-bit Instruction Cache Branch Prediction Hardware Multiply Hardware Divide
Memory Usage (e.g Stratix IV)	Two M9Ks (or equiv.)	Two M9Ks + cache

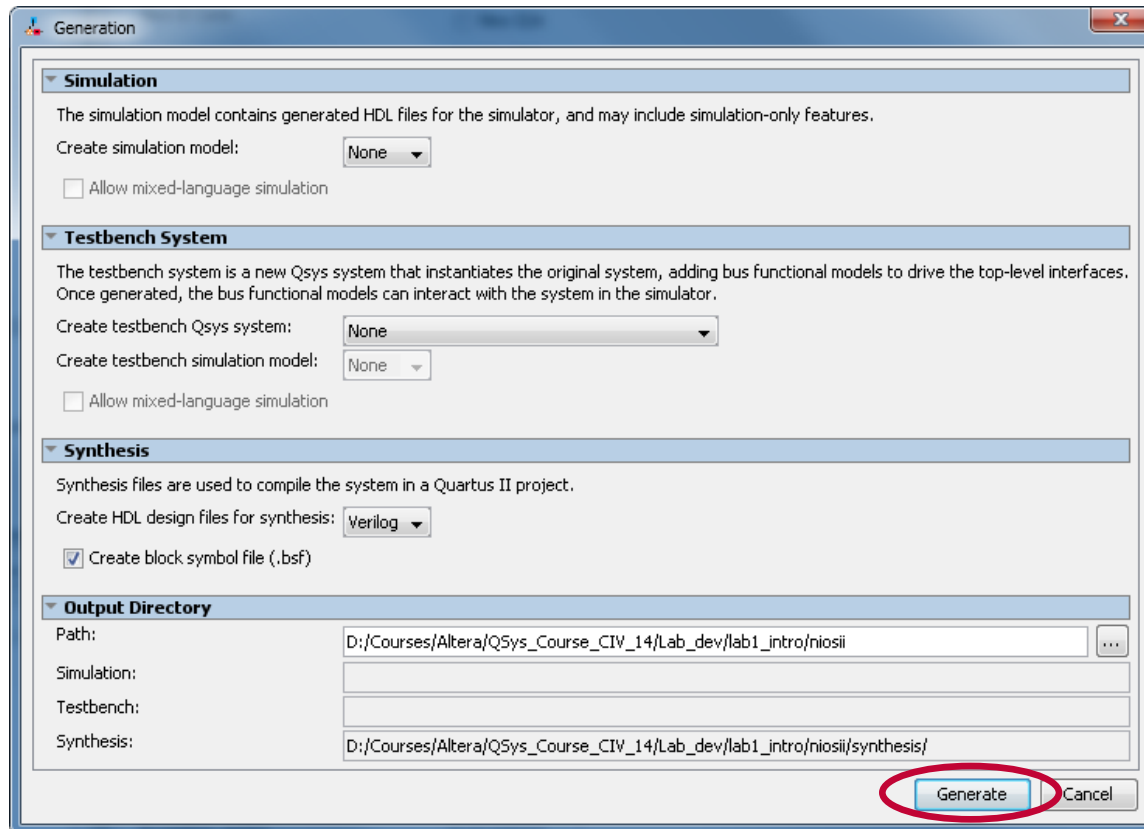
Меню Generate > HDL Example

- Генерирует шаблон создания экземпляра системы на Verilog или VHDL
- Можно использовать код для создания экземпляра QSys как модуля нижнего уровня в проекте



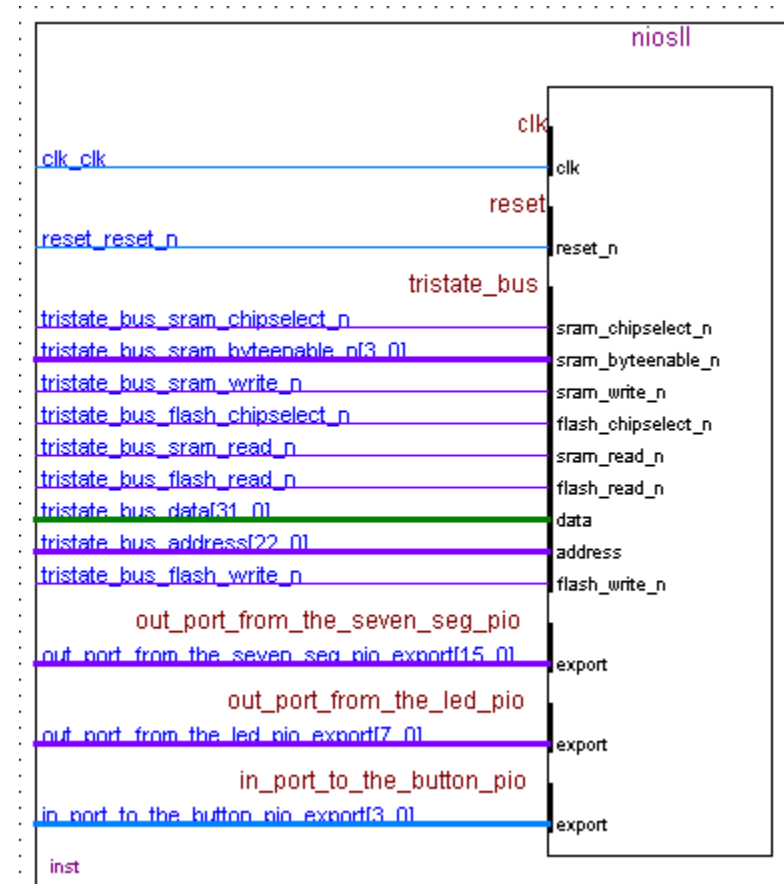
Меню Generate...

- Генерация исходных кодов для синтеза
- Возможность создания проекта и тестбенча для моделирования
- Возможность настройки путей для сохранения файлов
- **Проверьте предупреждения перед генерацией!**



Выходные файлы QSys

- **Verilog (.v) или SV(.sv)**
 - Исходные тексты для синтеза
- **Verilog (.v) или VHDL (.vhd)**
 - Исходные тексты и тестбенчи для моделирования
- **.BSF file**
 - Символ системы
- **.HTML file**
 - Отчет о генерации с описанием системы
- **.QIP file**
 - IP файл системы, надо добавить к проекту Quartus
- **.QSYS file**
 - Архив с описанием содержимого системы для последующего редактирования
- **.SOPCinfo file**
 - Шаблон проекта для средств разработки ПО

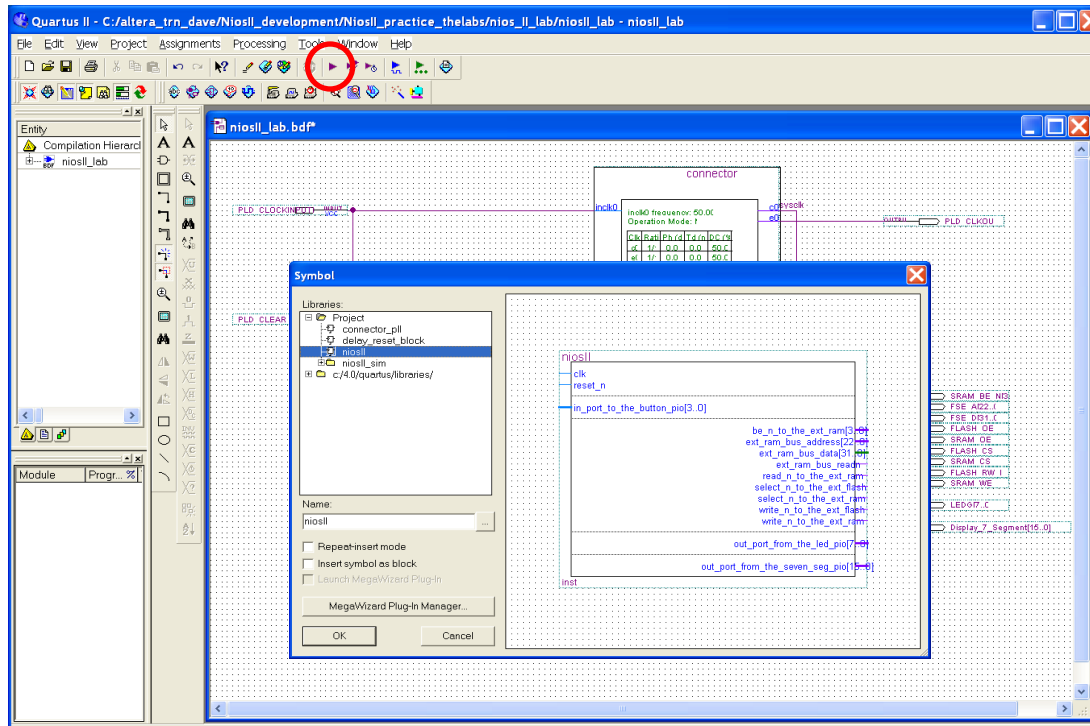


Файл .SOPCInfo

- Описывает состав системы на кристалле
 - Требуется для маршрута проектирования с использованием командной строки
 - Требуется для среды Nios II Software Build Tools for Eclipse
- Содержит следующую информацию
 - Имя проекта и версия QSys
 - Язык HDL
 - Имена и версии компонент в пути поиска
 - Размещение файлов на диске
 - Имена и версии модулей
 - Информация об их интерфейсах, включая имена и типы сигналов
 - Имена и значения параметров
 - Информация о каждом соединении, в то числе:
 - Какие компоненты соединяются
 - Базовый адрес, номер прерывания и т.д.
 - Карта памяти для каждого мастера (главного устройства)
- Формат XML




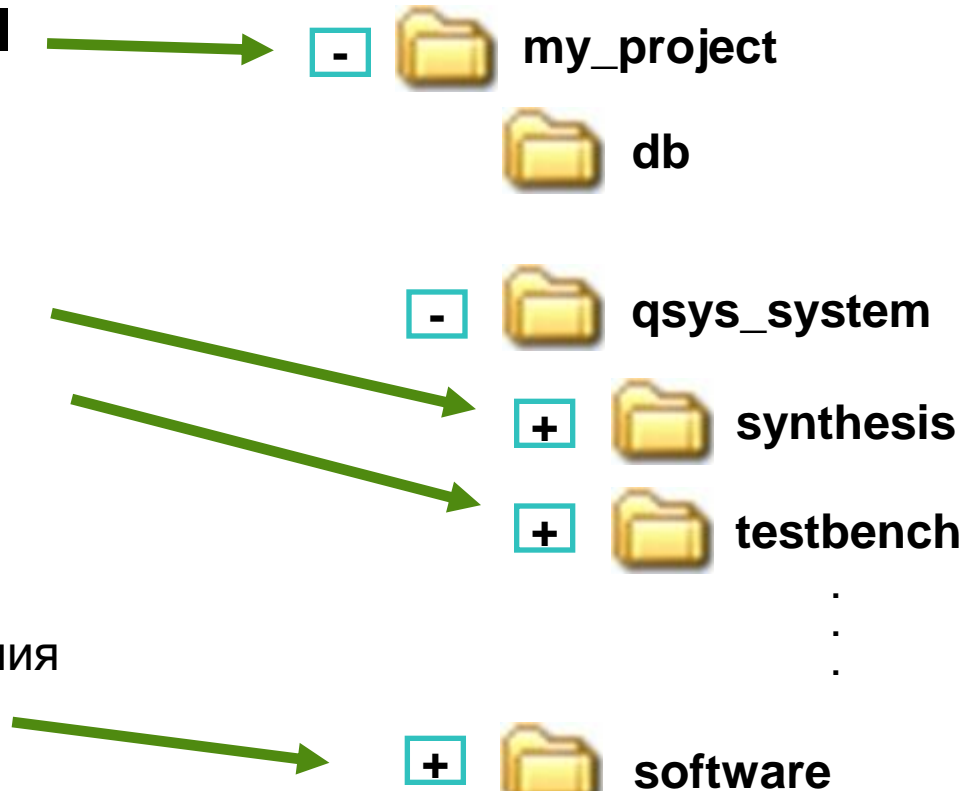








Схемный ввод

- Используйте блок (.bsf) файле проекта (.bdf)



- Добавьте файл **.QIP** к проекту
- Откомпилируйте проект в Quartus II

Каталог проекта Quartus II

- Папка проекта Quartus II →   my_project
 -  db
 - Папка с выходными файлами Qsys
 - Файлы для синтеза
 - Файлы для моделирования
 - Папка программного обеспечения
 - Исходные тексты приложения
 - Библиотеки
- 
-   qsys_system
 -   synthesis
 -   testbench
 - ⋮
 -   software

Настройка ядра Nios II в QSys

Добавление и конфигурация Nios II

Library

- Project
 - New Component...
 - System
- Library
 - Bridges
 - Bridges and Adapters
 - Clock and Reset
 - Configuration & Programming
 - DSP
 - Embedded Processors
 - Bitswap
 - Custom Instruction Interco
 - Custom Instruction Master
 - Custom Instruction Slave T
 - Floating Point Hardware
 - Floating Point Hardware 2
 - Hard Processor System
 - Nios II Processor**
 - Interface Protocols
 - Memories and Memory Controllers
 - Max10 Components

Nios II Processor - nios2_qsys_0

altera_nios2_qsys

Documentation

Core Nios II | Caches and Memory Interfaces | Advanced Features | MMU and MPU Settings | JTAG Debug Module

Select a Nios II Core

Nios II Core:

- ☐ Nios II/e
- ☐ Nios II/s
- ☒ Nios II/f

	Nios II/e	Nios II/s	Nios II/f
Nios II Selector Guide	RISC 32-bit	RISC 32-bit Instruction Cache Branch Prediction Hardware Multiply Hardware Divide	RISC 32-bit Instruction Cache Branch Prediction Hardware Multiply Hardware Divide Barrel Shifter Data Cache Dynamic Branch Prediction
Memory Usage (e.g Stratix IV)	Two M9Ks (or equiv.)	Two M9Ks + cache	Three M9Ks + cache

Hardware Arithmetic Operation

Hardware multiplication type: Embedded Multipliers

☐ Hardware divide

Reset Vector

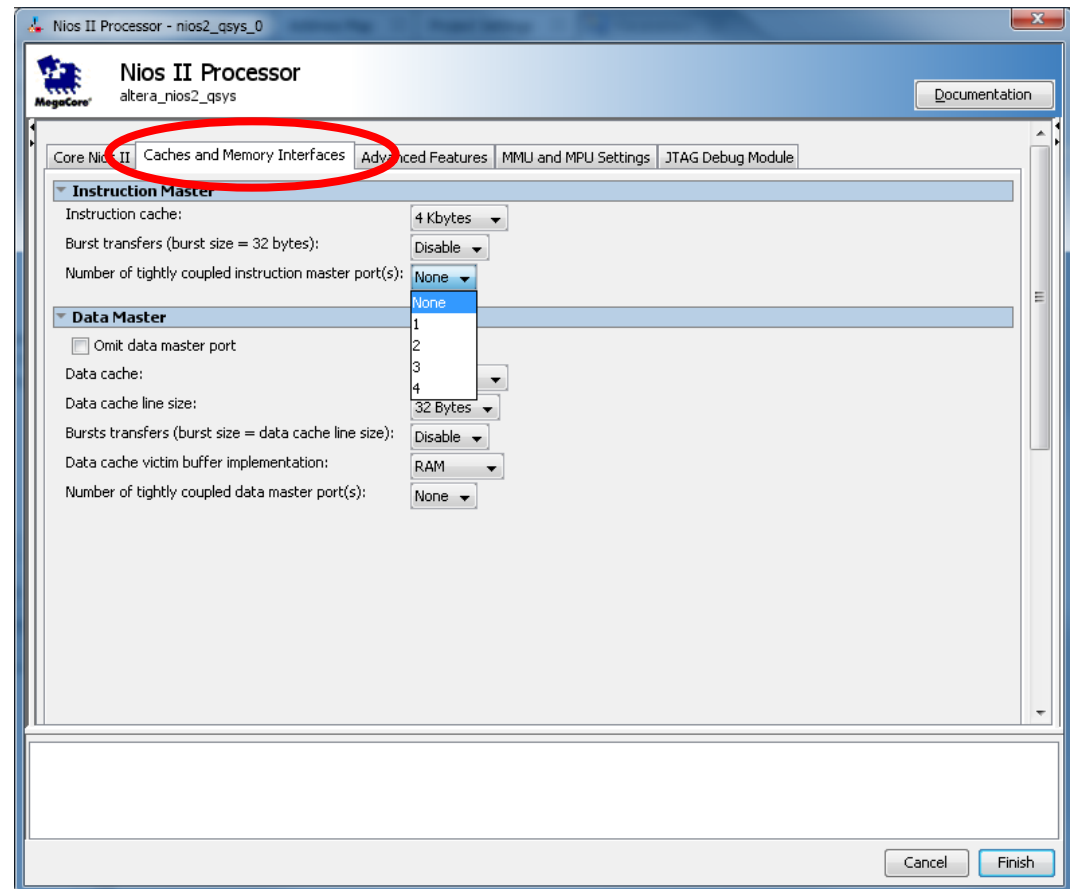
Cancel Finish

Адреса векторов **Reset** и **Exception**

- Устанавливаются *после* того, как модули памяти были добавлены в систему
- Вектор сброса **Reset** должен указывать на энергонезависимую память или встроенные блоки памяти, конфигурируемые при загрузке системы
 - При работе без загрузочного кабеля
- Вектор исключений **Exception** обрабатывает все исключения
 - Код обработчика включается в пакет поддержки платы
 - Поддерживаемые типы исключений
 - Программные исключения
 - Программные ловушки
 - Неподдерживаемые инструкции
 - Позволяет обеспечить совместимость версий Nios II
 - Аппаратные прерывания
 - Поддерживается 32 прерывания, чувствительных по уровню
 - Внешний контроллер прерываний предоставляет возможность неограниченного расширения

Настройка кэша и TCM

- Размер кэша команд и данных
 - В новой версии можно отключать кэш данных в версии Fast
 - При использовании TCM можно также отключить кэш команд
- Поддержка модулей тесно связанной памяти
 - память с быстрым доступом для данных и команд
- Настройка длин строки кэша данных
 - До 32 байт в строке кэша для повышения скорости обмена с внешней SDRAM за счет увеличения burst size



TCM (Tightly Coupled Masters)

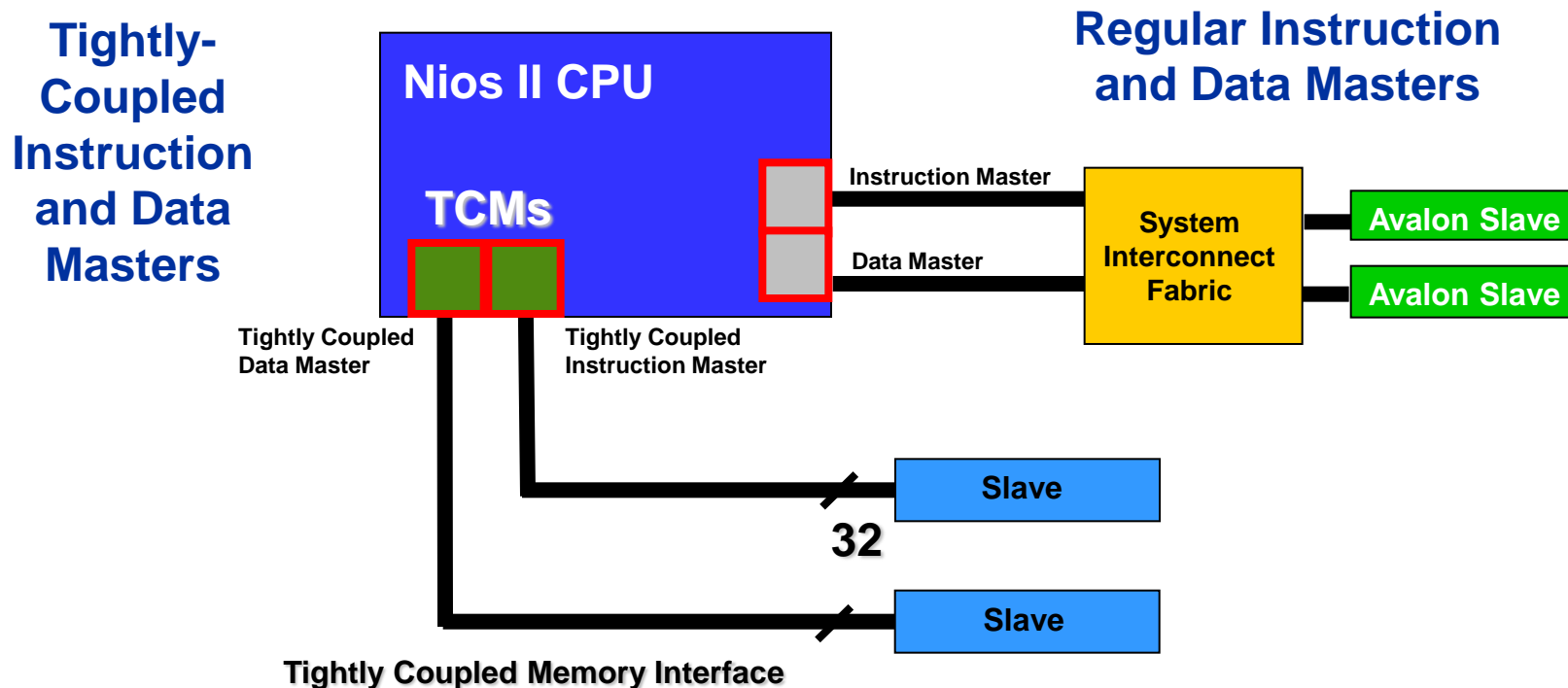
- Обеспечивает быстрый доступ Nios II к памяти на кристалле
- Добавляет “локальные” интерфейсы, в которых CPU является задатчиком (Master Interfaces)

	TCM команд	TCM данных
Nios II/f	До 4	До 4
Nios II/s	До 4	Нет
Nios II/e	Нет	Нет

- Допускают присоединение только некоторых типов “быстрых” периферийных модулей
 - Т.е. памяти на кристалле
 - Задержка чтения 1, задержка записи 0
 - Для повышения скорости обмена используются двухпортовые модули

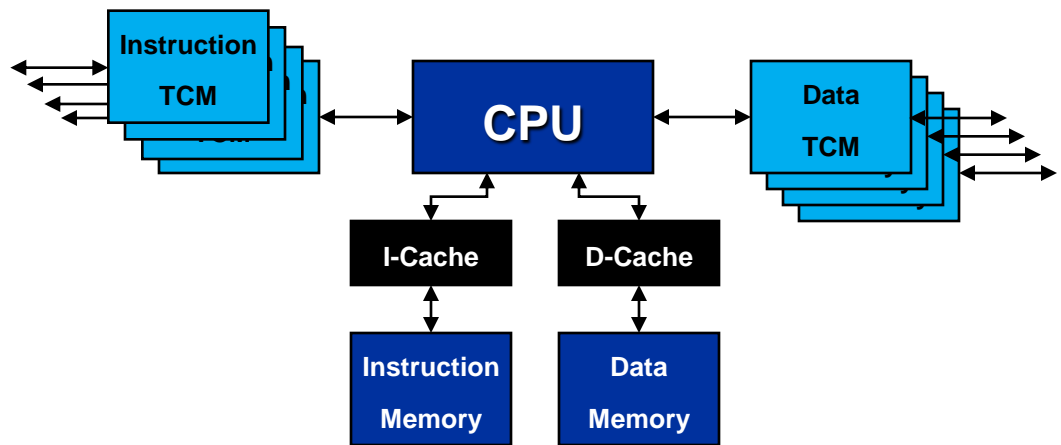
TCM (Tightly Coupled Masters)

- Связь с подчиненными устройствами через Tightly Coupled Memory Interfaces не требует коммутации
- Двухпортовые модули памяти могут реализовывать одновременно интерфейс TCM slave и Avalon slave



ТСМ и кэш

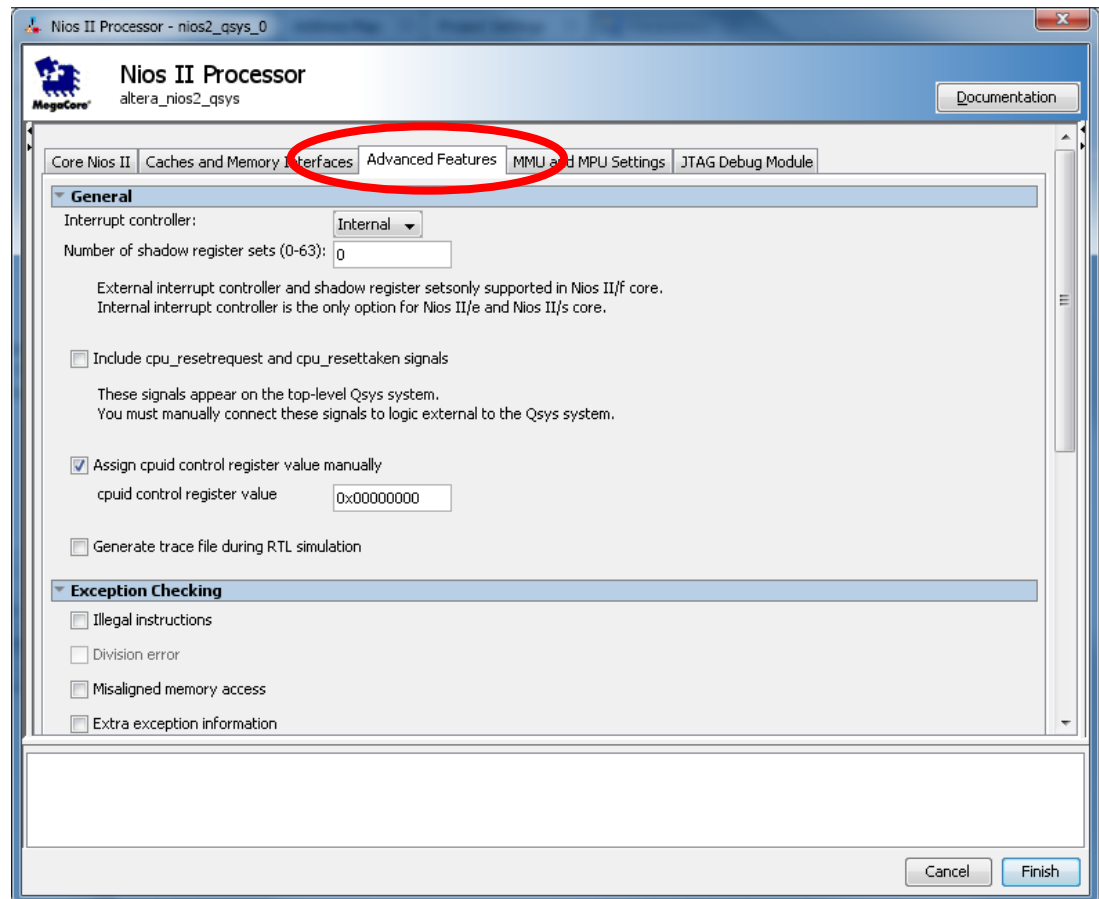
- Доступ к модулям тесно связанной памяти не кэшируется
 - Работают как кэш с вероятностью попадания 100%
 - Рекомендуется для размещения обработчиков прерываний и других критичных к времени выполнения функций
- Отображаются в карту памяти системы, режим доступа определяется декодером адреса процессора



- Рекомендуется размещать ТСМ в верхних адресах
 - Упрощает логику декодирования в декодере адреса
 - Повышает производительность

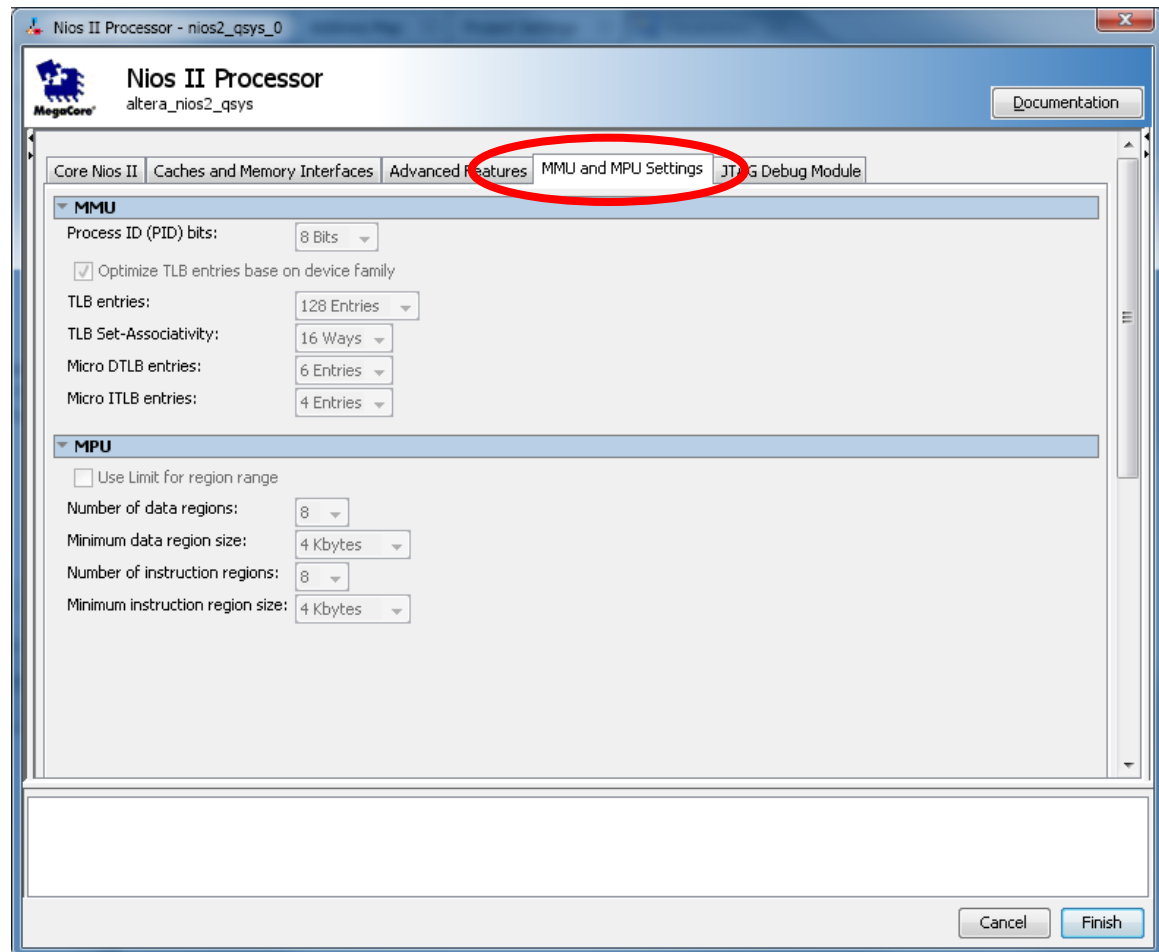
Закладка Advanced Features

- Выбор внутреннего или внешнего контроллера прерываний
- Вывод сигналов **cpu_resetrequest** и **cpu_resettaken**
- Ручная настройка **cpuid**
- Поддержка исключений



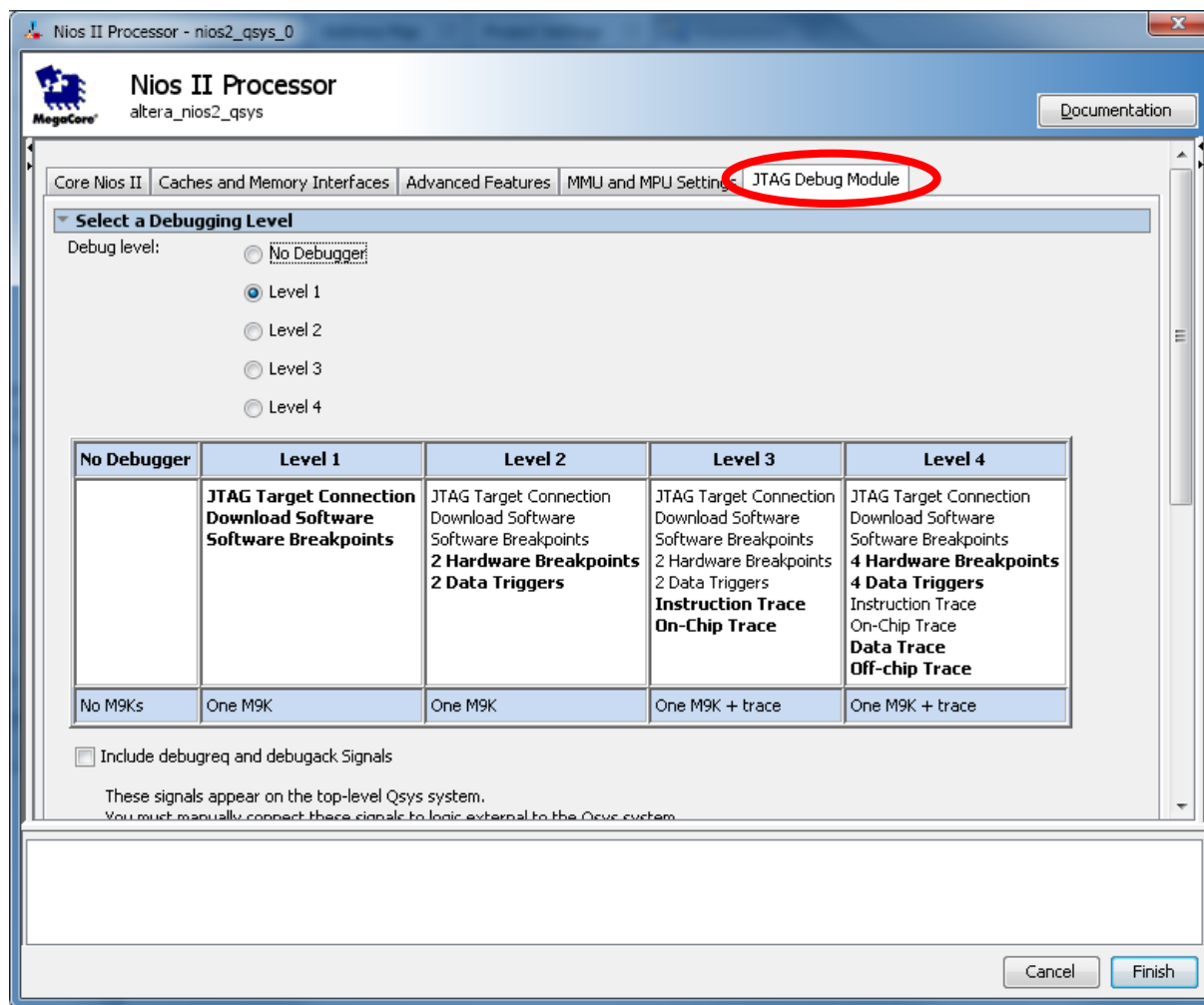
Закладка MMU and MPU Settings

- Настройка MMU и MPU в соответствии с требованиями приложения или ОС
- MMU и MPU – взаимоисключающие модули

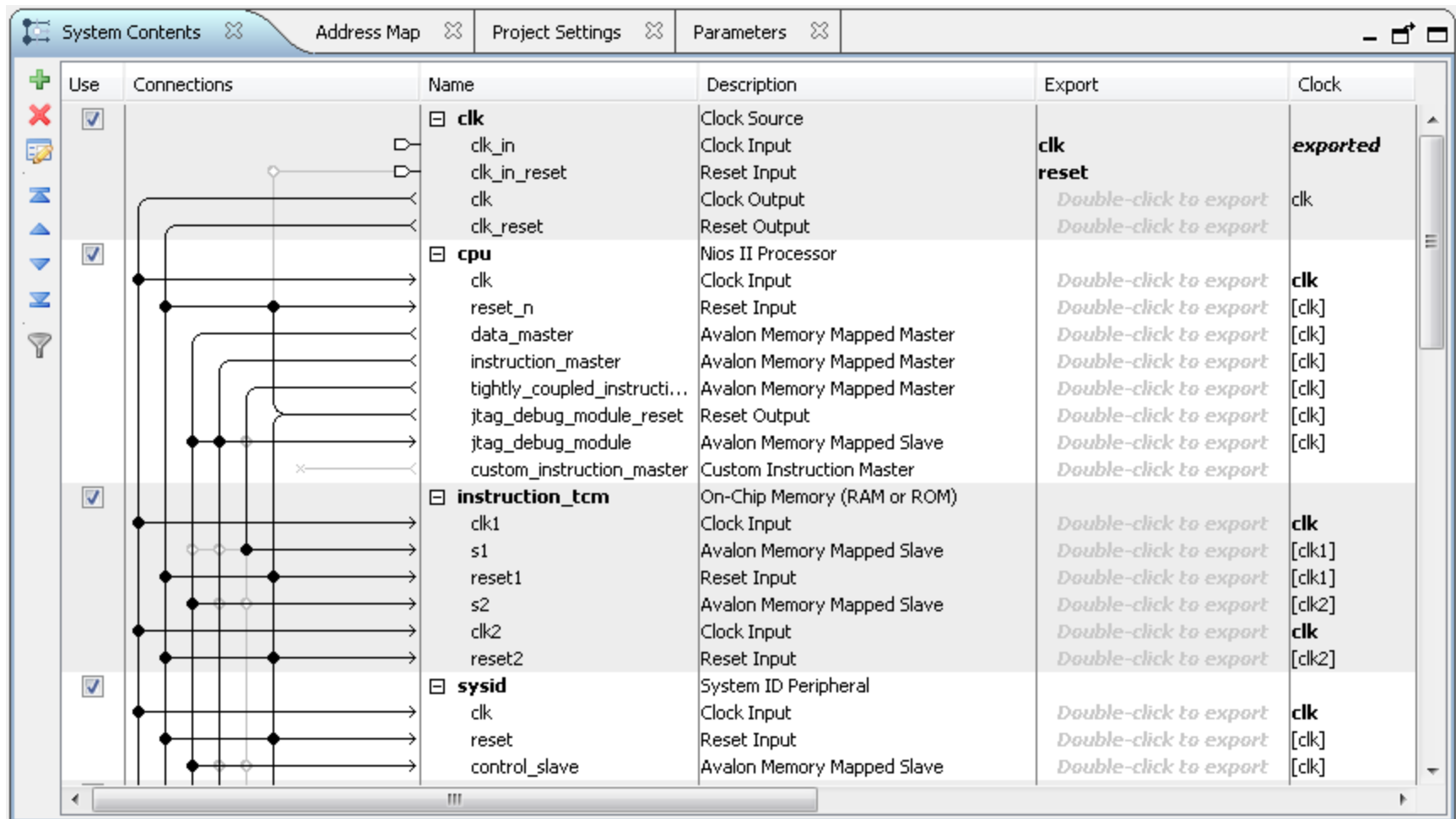


Закладка JTAG Debug Module

- Выбор возможностей внутрисхемной отладки через JTAG



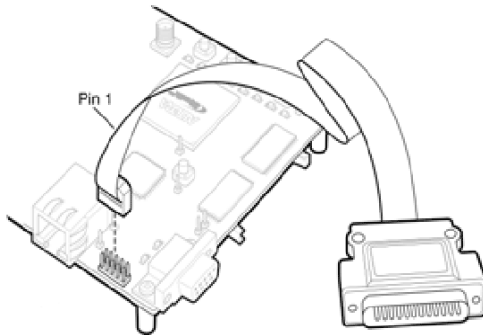
CPU в составе системы



Вспомогательные периферийные модули

■ Модуль JTAG UART

- Реализует:
 - Конфигурацию устройства
 - Программирование Flash
 - Загрузку кода
 - Отладку
 - Консольный вывод через STDIO



■ Модуль System ID

- Используется для синхронизации версий аппаратного и программного обеспечения
- Периферийный модуль с двумя регистрами на чтение
 - Регистр 1 – Идентификатор, задается пользователем
 - Регистр 2 – время и дата генерации системы в QSys
- Может использоваться для проверки соответствия загружаемого программного кода конфигурации аппаратуры

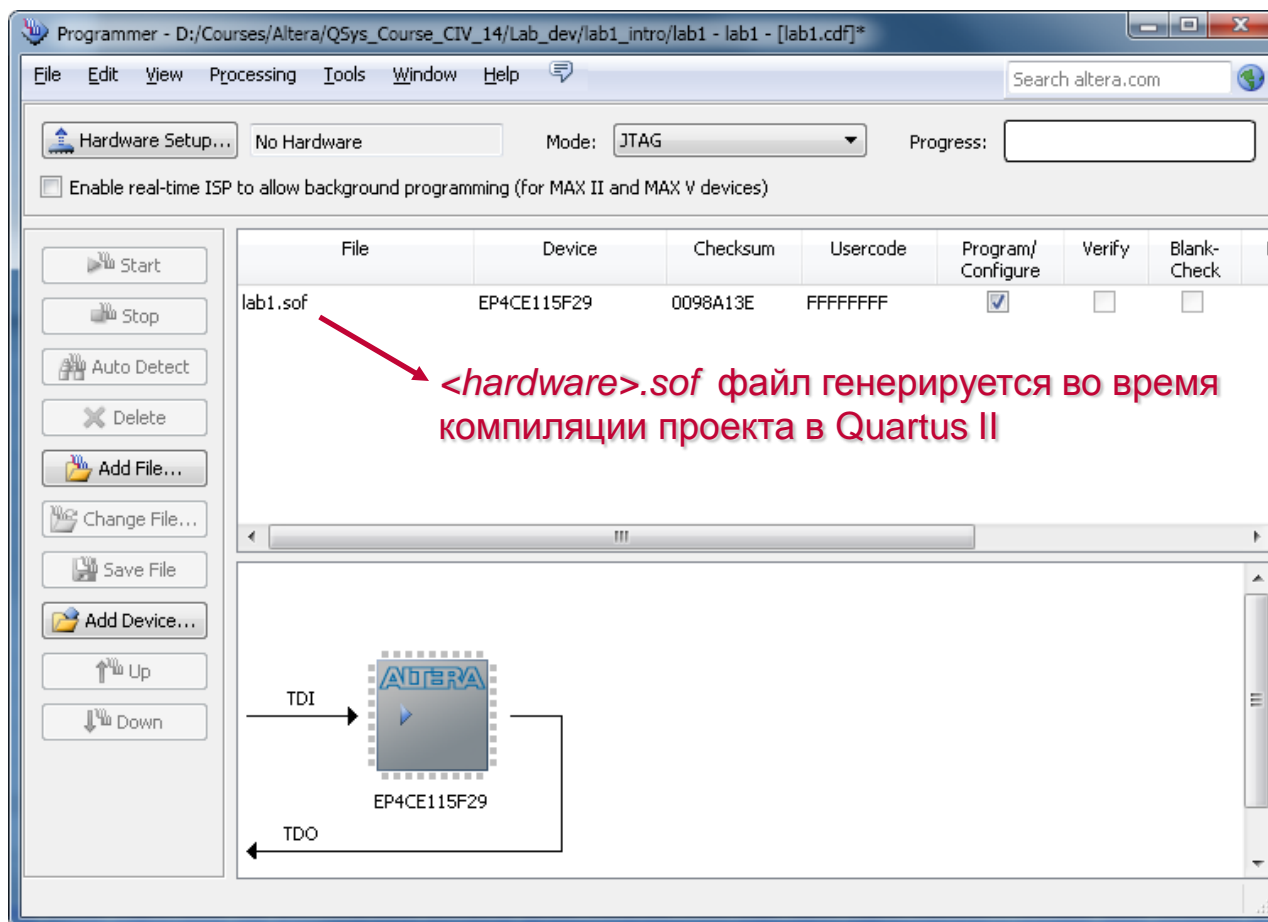
Во всех проектах с Nios II
должен присутствовать
модуль **system ID**

ALTERA®

Конфигурация ПЛИС

Конфигурация из программатора Quartus II

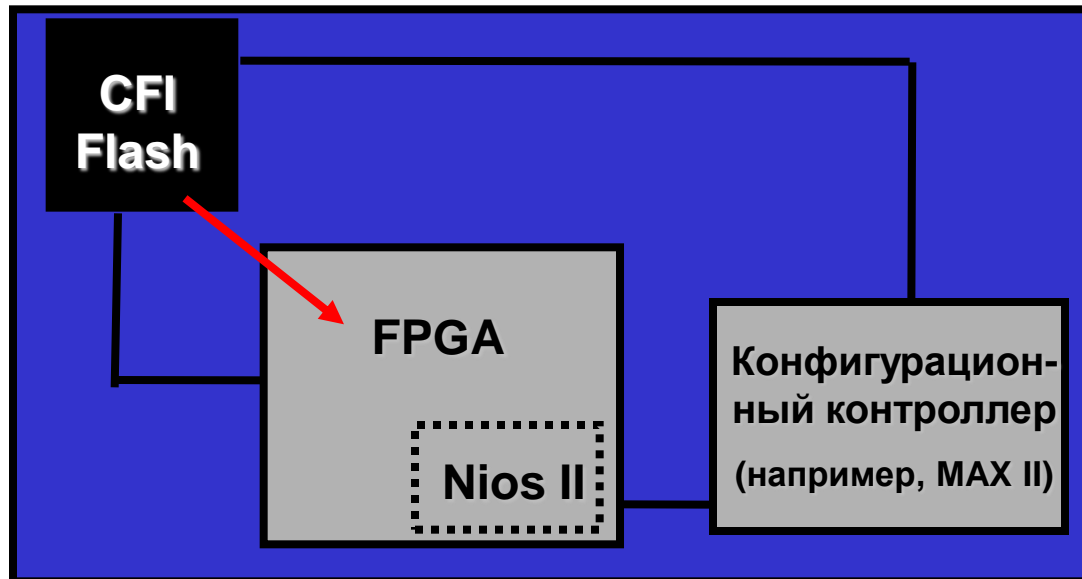
- Запускается из Quartus II или Nios II SBT



Конфигурация из Flash

- Используются микросхема энергонезависимой памяти и конфигурационный контроллер
 - Конфигурационный контроллер считывает данные из Flash памяти
 - В Flash памяти должен содержаться образ прошивки ПЛИС
 - Также в Flash памяти может храниться код Nios II
 - Средства разработки предоставляют возможность реализовать программирование Flash через ПЛИС

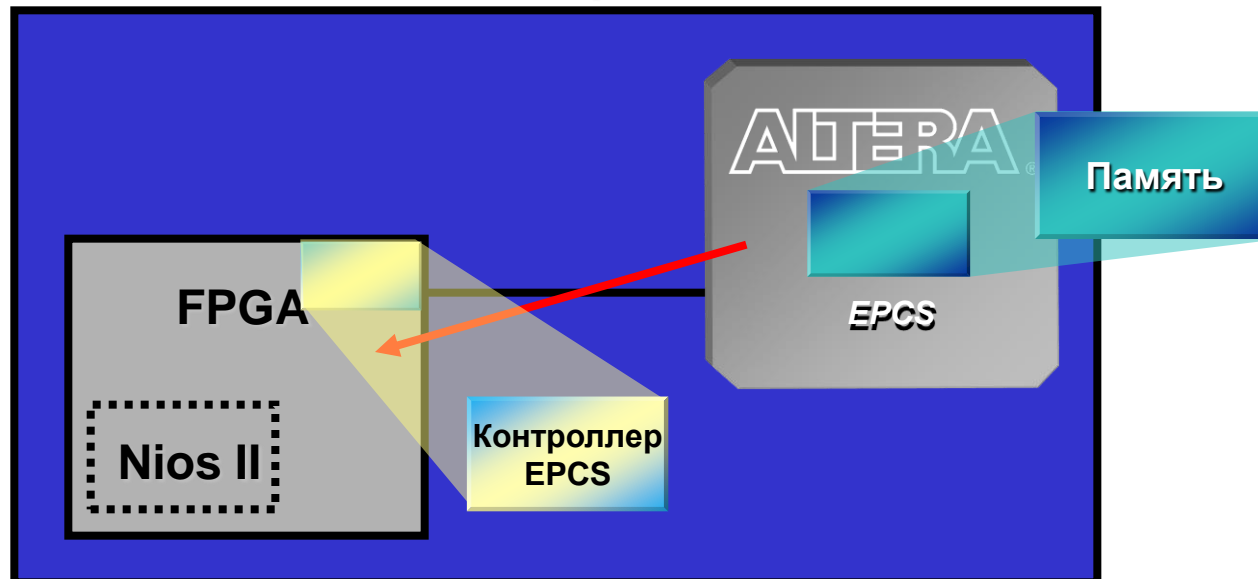
Nios II Development Board



Конфигурация из EPCS

- Одно устройство, низкая стоимость
 - Хранит конфигурацию
 - Возможно хранение кода
 - Низкая стоимость
 - Средства разработки предоставляют возможность реализовать программирование Flash через ПЛИС

Nios II Development Board



Лабораторная работа 1

Создание системы в Qsys