Определение системы в SOPC Builder

Вы используете SOPC Builder для определения характеристик аппаратной части системы Nios II, например, какое ядро Nios II использовать, и какие компоненты входят в систему. SOPC Builder не определяет поведение программной части, например, какая память используется для хранения инструкций, или как передать поток символов *stderr*.

В этой секции вы выполните следующие пункты:

1. Выбор FPGA и настроек тактового сигнала.
2. Добавление ядра Nios II, внутри чиповой памяти и прочих компонентов.
3. Определение базового адреса и приоритета запроса прерываний (IRQ).
4. Генерирование системы SOPC Builder.

Процесс разработки SOPC Builder не обязательно линеен. Пошаговое проектирование в этом учебном пособии представляет общий прямолинейный порядок, понятный начинающим пользователям. Таким образом, вы можете выполнять пункты проектирования системы SOPC Builder в любом порядке.

Выбор FPGA и настроек тактового сигнала

Секции **Исполнитель** и **Настройки тактового сигнала** на вкладке **Содержание системы** определяют взаимодействие системы SOPC Builder с другими устройствами в системе. Выполните следующие пункты:

1. Выберите **Семейство чипов** из производимых Altera FPGA, которое вы будете использовать.

Если появляется предупреждение, что выбранное семейство чипов не существует в проекте Quartus, проигнорируйте это предупреждение. Вы можете выбрать чип позже в настройках проекта Quartus.

2. В документации на вашу плату, обратите внимание на частоту сигнала тактового генератора, поступающую на FPGA.

Руководство пользователя платой разработчика Altera находится на странице Литература: Наборы разработчика на веб-сайте Altera.

3. Дважды кликните на тактовую частоту в столбце **МГц** для *clk_0*. В системе SOPC Builder *clk_0* – это имя тактового сигнала по умолчанию. Частота, введённая вами для *clk_0,* должна соответствовать частоте, поступающей от генератора на FPGA.
4. Введите частоту и нажмите **Ввод**.

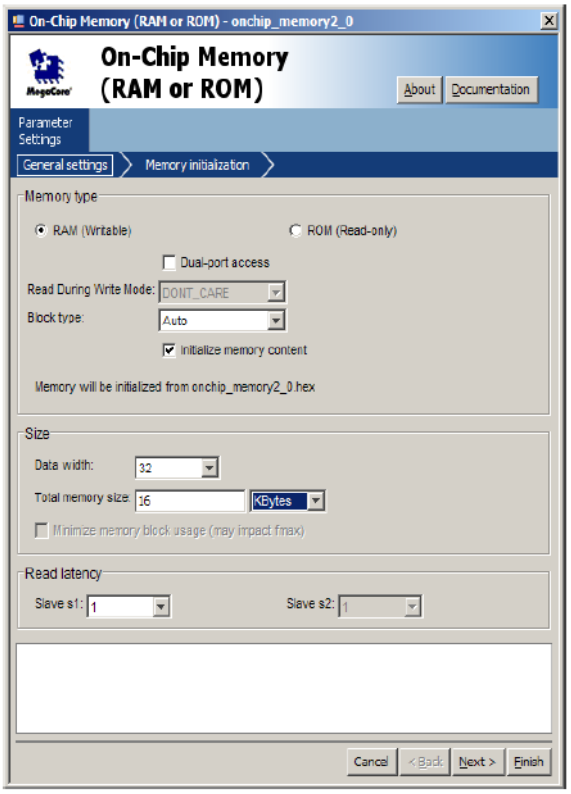
Далее вы начнёте добавлять компоненты в систему SOPC Builder. После добавления каждого компонента, вы можете его сконфигурировать в соответствии со спецификацией проекта.

Добавление внутри чиповой памяти

Процессорным системам требуется не менее одной памяти под данные и инструкции. Этот пример проектирования использует одну память на чипе 20 Кб под данные и инструкции. Для добавления памяти выполните следующие пункты:

1. В списке доступных компонентов (с левой стороны на вкладке **Содержимое системы**), раскройте **Память и контроллеры памяти**, раскройте **На чипе**, затем кликните на **Внутри чиповая память (RAM или ROM).**
2. Кликните **Добавить**. Раскроется интерфейс MegaWizard для внутри чиповой памяти (RAM или ROM). На рисунке 1-5 показан его графический интерфейс.
3. В списке **Тип блока** выберите **Автоматически**.
4. На панели **Общий размер памяти**, введите 20 и выберите **Килобайт**, чтобы определить память размером 20 Кб.
5. Не меняйте никакую настройку по умолчанию.

Figure 1–5. On-Chip Memory MegaWizard



6. Кликните **Финиш**. Вы вернётесь на вкладку **Содержимое системы** SOPC Builder, а внутри чиповая память сразу появится в таблице доступных компонентов.

Чтобы подробнее изучить внутри чиповую память, вы можете кликнуть на **Документацию** в интерфейсе MegaWizard для внутри чиповой памяти (RAM или ROM). Такая документация доступна для каждого компонента в интерфейсе MegaWizard/

7. Правым кликом на внутри чиповую память выберите **Переименовать**.
8. Введите *onchip_mem* и нажмите **Ввод**.

Вам необходимо точно написать имя этого учебного компонента. Иначе, учебная программа, написанная для этой системы Nios II, в дальнейшем будет сбоить. Главное, хорошим стилем является давать описательные имена аппаратным компонентам. Программы Nios II используют эти символьные имена для доступа к аппаратным компонентам. Поэтому выбор вами имён компонентов делает наглядной и читабельной программу Nios II.

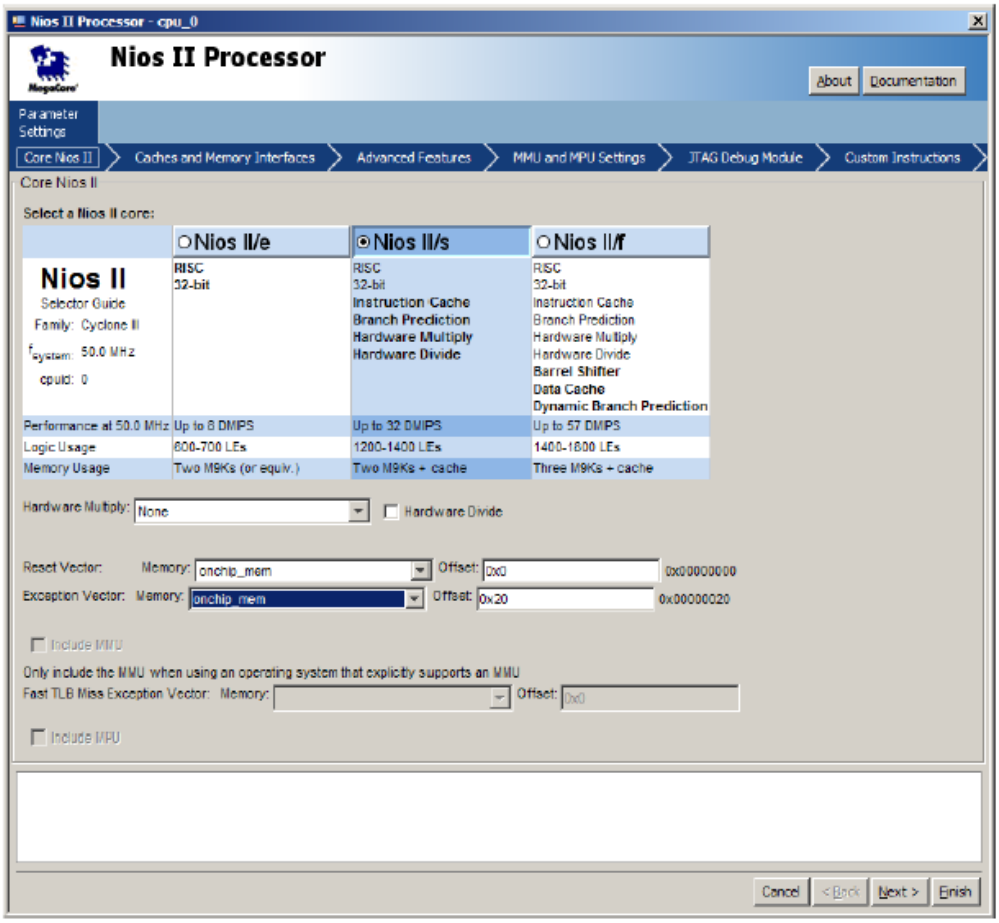
Добавление ядра процессора Nios II

В этой секции вы добавите ядро Nios II/s и сконфигурируете его для использования 2 Кб внутри чиповой памяти под кэш инструкций. По замыслу обучения, пример учебного проекта использует "стандартное" ядро Nios II/s, которое является сбалансированным компромиссом между характеристиками и использованием ресурсов. Реально, ядро Nios II/s более мощное, чем требуется для большинства контрольных приложений.

Выполните следующие пункты для добавления ядра Nios II/s в вашу систему:

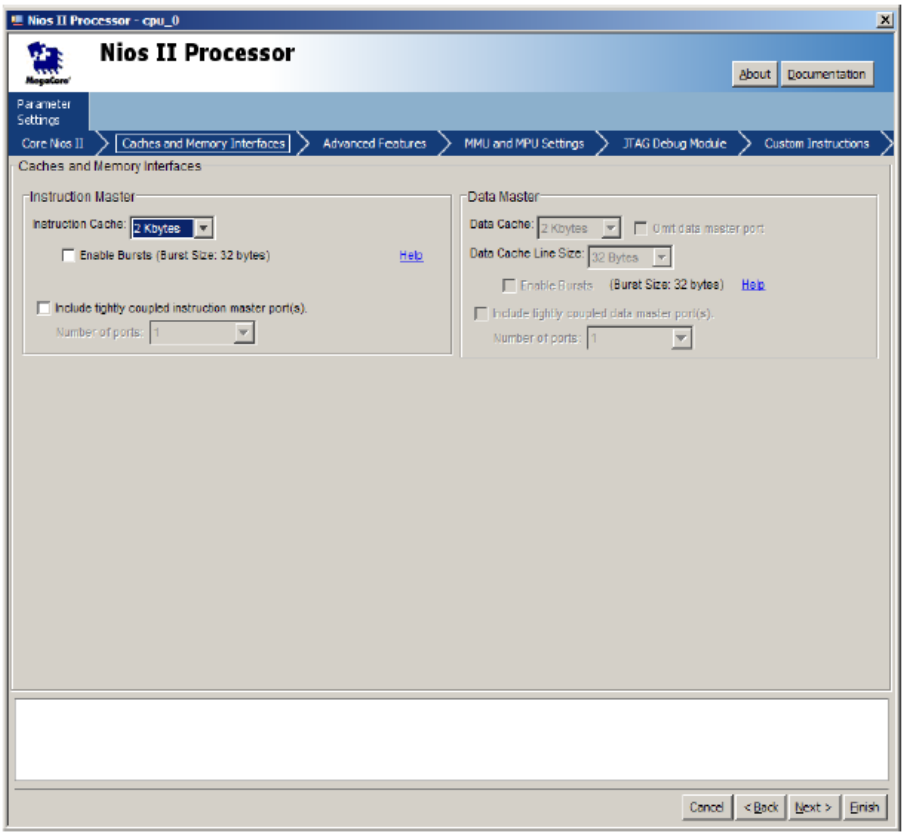
1. В списке доступных компонентов, раскройте **Процессоры**, затем кликните **Процессоры Nios II**.
2. Кликните **Добавить**. Раскроется интерфейс MegaWizard для процессоров Nios II, отображающий страницу **Ядра Nios II**. На рисунке 1-6 показана графическая оболочка.
3. Под **Выбрать ядро Nios II** выберите **Nios II/s**.
4. В списке **Аппаратное умножение** выберите **Нет**.
5. Выключите **Аппаратное деление**.
6. Под **Вектором сброса** выберите **onchip_mem** в списке **Память** и введите 0x0 в панели **Начальный номер (Офсет)**.
7. Под **Вектором исключения** выберите **onchip_mem** в списке **Память** и введите 0x20 в панели **Начальный номер (Офсет)**.

Figure 1–6. Nios II MegaWizard – Nios II Core Page



8. Кликните **Интерфейсы с памятью и кэшем**. Раскроется страница **Интерфейсы с памятью и кэшем**. На рисунке 1-7 показан его графический интерфейс.
9. В списке **Кэш инструкций** выберите **2 Кбайта**.
10. Выключите **Разрешить пакеты**.
11. Выключите **Содержит прочно сопряжённые инструкции для мастер порта (-ов)**.

Figure 1–7. Nios II MegaWizard – Caches and Memory Interfaces page

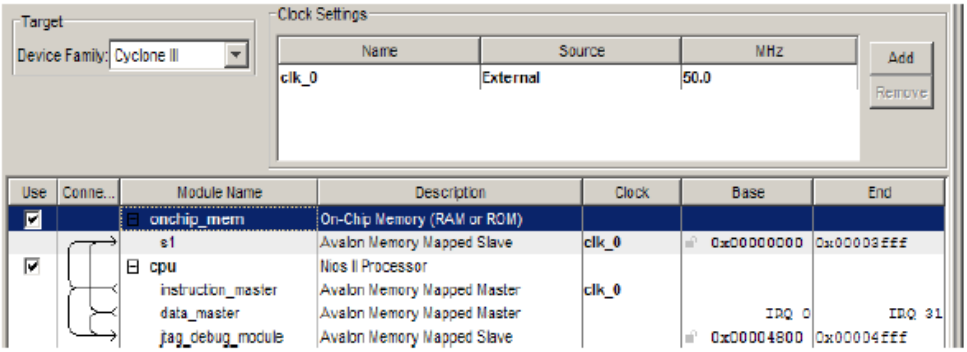


12. Не меняйте никакие настройки на страницах **Расширенные средства, Настройки MMU и MPU, Модуль отладки JTAG** и **Собственные инструкции**.
13. Кликните **Финиш**. Вы вернётесь на вкладку **Содержимое системы** SOPC Builder, а модуль ядра Nios II сразу появится в таблице доступных компонентов.

За дополнительной информацией о конфигурировании ядра Nios II, обратитесь к главе "*Инсталляция процессора Nios II в SOPC Builder*" в *Настольной книге процессора Nios II*".

SOPC Builder автоматически подключит мастер порты инструкций и данных ядра Nios II к ведомому порту памяти. На рисунке 1-8 показан графический интерфейс. Когда строится система, всегда контролируйте, что SOPC Builder автоматически подключает в соответствии с требованиями вашей системы.

Figure 1–8. System Contents Tab with the Nios II Core and On-Chip Memory



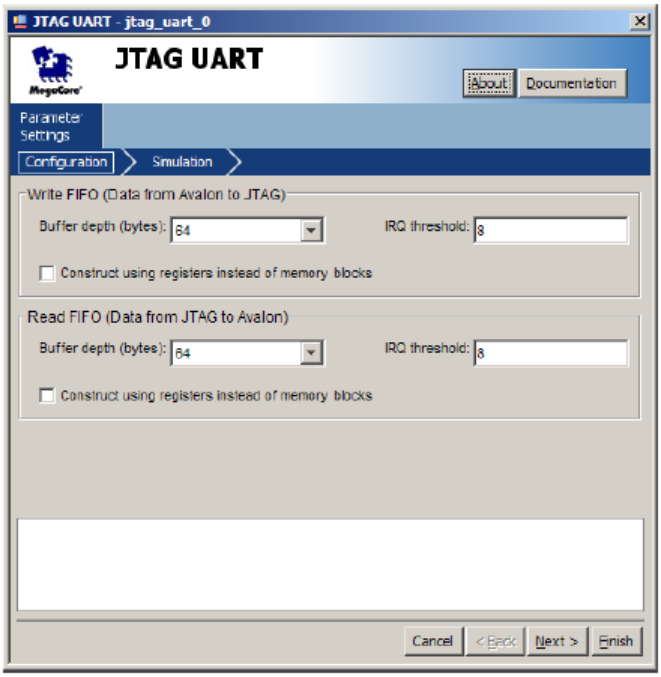
За дополнительной информацией о подключении памяти к системе Nios II, обратитесь к главе "*Создание подсистемы памяти с помощью SOPC Builder"* в томе 4 *Настольной книге процессора Quartus II"*.

Добавление JTAG UART

JTAG UART – это простой путь обмена символьными данными с процессором Nios II посредством загрузочного кабеля USB-Blaster. Выполните следующие пункты для добавления JTAG UART:

1. В списке доступных компонентов раскройте **Протоколы интерфейса**, затем **Последовательный**, и кликните **JTAG UART**.
2. Кликните **Добавить**. Раскроется интерфейс MegaWizard для JTAG UART. На рисунке 1-9 показан графический интерфейс.
3. Не меняйте настройки по умолчанию.

Figure 1–9. JTAG UART MegaWizard



4. Кликните **Финиш**. Вы вернётесь на вкладку **Содержимое системы** SOPC Builder, а модуль JTAG UART сразу появится в таблице доступных компонентов.

SOPC Builder автоматически подключит мастер порт данных ядра Nios II к ведомому порту JTAG UART. (Мастер порт инструкций не может подключаться к JTAG UART, поскольку JTAG UART – это не устройство памяти и не может следовать инструкциям процессора Nios II). Когда строится система, всегда контролируйте, что SOPC Builder автоматически подключает в соответствии с требованиями вашей системы.

За дополнительной информацией о JTAG UART, обратитесь к главе "*Ядро JTAG UART"* в томе 5 *Настольной книге процессора Quartus II"*.

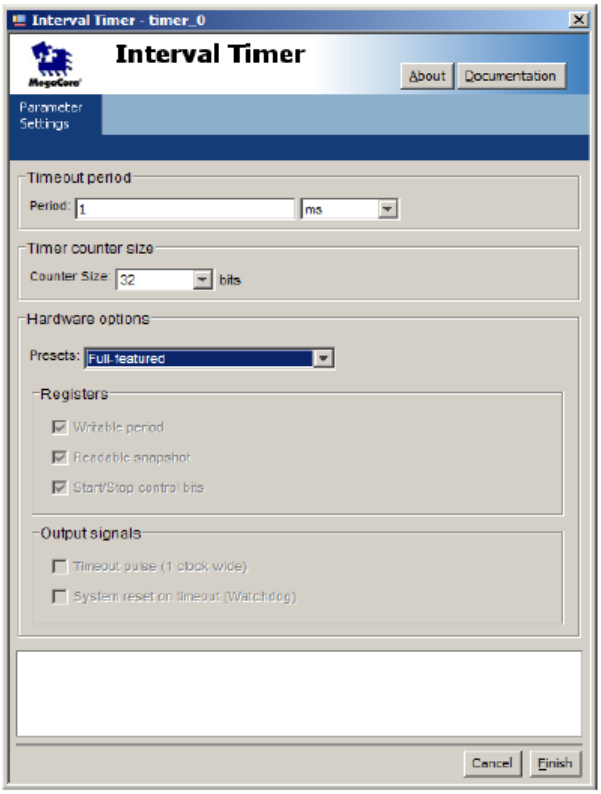
Добавление интервального таймера

Большинство контрольных систем используют компонент таймера, чтобы разрешить точный подсчёт времени. Чтобы создать периодический такт системных часов, Nios II HAL обращается к таймеру.

Выполните следующие пункты для добавления таймера:

1. В списке доступных компонентов раскройте **Периферия**, затем **Периферия микроконтроллера**, и кликните **Интервальный таймер**.
2. Кликните **Добавить**. Раскроется интерфейс MegaWizard для интервального таймера. На рисунке 1-10 показан графический интерфейс.
3. В списке **Предустановка**, выберите **Полнофункциональный**.
4. Не меняйте настройки по умолчанию.

Figure 1–10. Interval Timer MegaWizard



5. Кликните **Финиш**. Вы вернётесь на вкладку **Содержимое системы** SOPC Builder, а модуль интервального таймера сразу появится в таблице доступных компонентов.
6. Правым кликом на интервальном таймере выберите **Переименовать**.
7. Введите имя *sys_clk_timer* и нажмите **Ввод**.

За дополнительной информацией о таймере, обратитесь к главе "*Ядро таймера*" в томе 5 *Настольной книге процессора Quartus II*".

Добавление периферии идентификатора системы

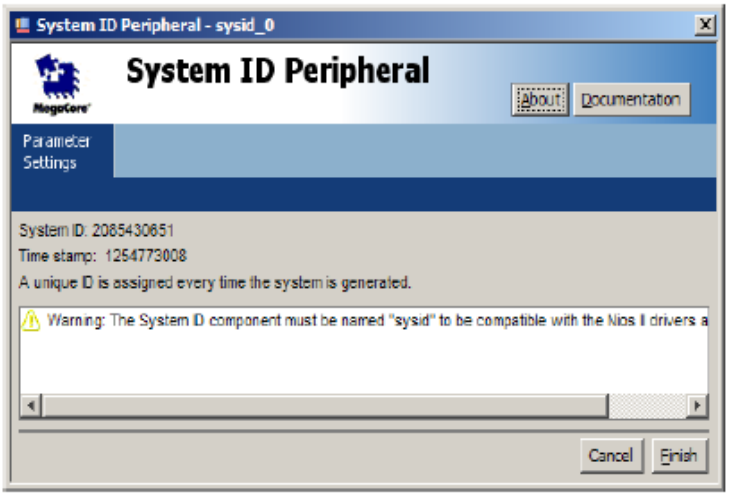
Периферия идентификатора системы защищает от случайной загрузки программы, скомпилированной для другой системы Nios II. Если в системе присутствует периферия идентификатора системы, инструмент создания программы Nios II под Eclipse предотвращает загрузку программы, скомпилированной для другой системы.

Выполните следующие пункты для добавления периферии идентификатора системы:

1. В списке доступных компонентов раскройте **Периферия**, затем **Отладка и Исполнение**, и кликните **Периферия идентификатора системы**.

2. Кликните **Добавить**. Раскроется интерфейс MegaWizard для периферии идентификатора системы. На рисунке 1-11 показан графический интерфейс. Периферия идентификатора системы не имеет конфигурируемых пользователем настроек.

Figure 1–11. System ID Peripheral MegaWizard



3. Кликните **Финиш**. Вы вернётесь на вкладку **Содержимое системы** SOPC Builder, а модуль периферии идентификатора системы сразу появится в таблице доступных компонентов.
4. Правым кликом на периферии идентификатора системы выберите **Переименовать**.
5. Введите имя *sysid* и нажмите **Ввод**.

За дополнительной информацией о периферии идентификатора системы, обратитесь к главе "*Ядро периферии идентификатора системы*" в томе 5 *Настольной книге процессора Quartus II*".

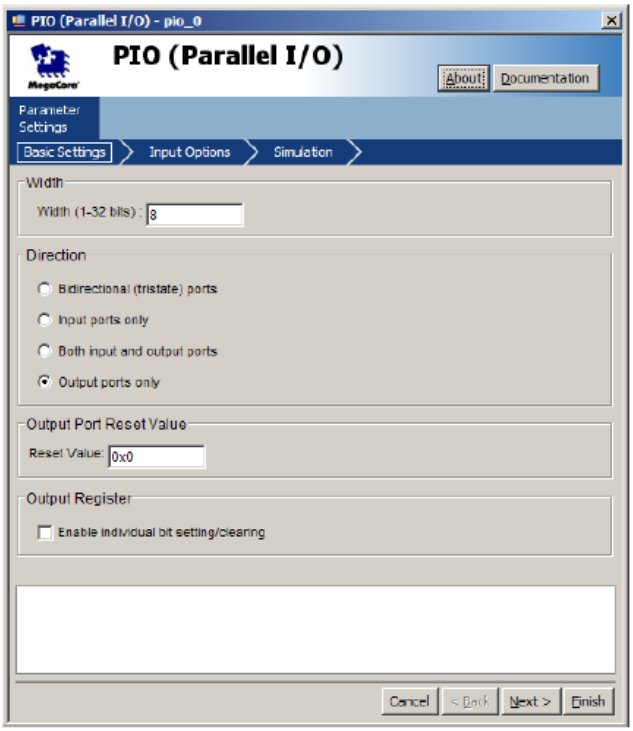
Добавление PIO

Сигналы PIO – это простой способ для процессорной системы Nios II принимать входные стимулы и выводить выходные сигналы. Комплексные контрольные приложения могут использовать сотни сигналов PIO, которые может наблюдать процессор Nios II. В этом примере проекта используется восемь сигналов PIO, ведущих на светодиоды на плате.

Выполните следующие пункты для добавления PIO. Выполните эти пункты даже, если на вашей плате нет светодиодов:

1. В списке доступных компонентов раскройте **Периферия**, затем **Периферия микроконтроллера**, и кликните **PIO (Параллельные I/O)**.
2. Кликните **Добавить**. Раскроется интерфейс MegaWizard для PIO (Параллельных I/O). На рисунке 1-12 показан графический интерфейс.
3. Не меняйте настройки по умолчанию. Интерфейс MegaWizard устанавливает по умолчанию 8-ми битные выходы PIO, что полностью соответствует потребностям примера проекта.

Figure 1–12. PIO MegaWizard



4. Кликните **Финиш**. Вы вернётесь на вкладку **Содержимое системы** SOPC Builder, а модуль PIO сразу появится в таблице доступных компонентов.
5. Правым кликом на PIO выберите **Переименовать**.
6. Введите имя *led_pio* и нажмите **Ввод**.

За дополнительной информацией о периферии идентификатора системы, обратитесь к главе "*Ядро PIO"* в томе 5 *Настольной книге процессора Quartus II"*.

Определение базового адреса и приоритета запроса прерываний (IRQ)

К этому моменту, вы добавили все необходимые аппаратные компоненты в систему. Теперь вам нужно определить, форму взаимодействия этих компонентов в системе. В этой секции вы назначите базовый адрес для каждого ведомого компонента, а также назначите приоритет запроса прерываний (IRQ) для JTAG UART и интервального таймера.

SOPC Builder позволяет легко назначить базовые адреса с помощью команды **Авто назначение базовых адресов**. Для большинства систем, включая этот проект, допускается использование команды **Авто назначение базовых адресов**. Однако, вы, по необходимости, можете скорректировать базовые адреса. Следуйте следующим указаниям для назначения базовых адресов:

■ Ядра процессора Nios II имеют диапазон адреса в 31 бит. Вы можете назначить базовый адрес между 0x00000000 и 0x7FFFFFFF.

■ Программы для Nios II используют символьные константы для обращения к адресу. Не беспокойтесь о выборе значения адреса, потому что его просто запомнить.

■ Значения адреса разных компонентов, различающихся только одним битом адреса, способствуют созданию более эффективных устройств. Но не пытайтесь укомплектовать все базовые адреса в наименьший диапазон адресов, поскольку это уменьшает эффективность устройств.

■ SOPC Builder не пытается выстроить по порядку различные компоненты памяти в непрерывном поле адреса. Например, если вы хотите создать непрерывное поле адреса для внутри чиповой RAM и внешней RAM, вы должны однозначно задать базовые адреса.

SOPC Builder имеет команду **Авто назначение IRQ**, которая позволяет подключить IRQ сигналы для получения правильных аппаратных данных. Однако, эффективное назначение IRQ требует понимание того, как программа обращается к ним. Поскольку SOPC Builder не управляет поведением программы, он не сможет предположить грамотное назначение IRQ.

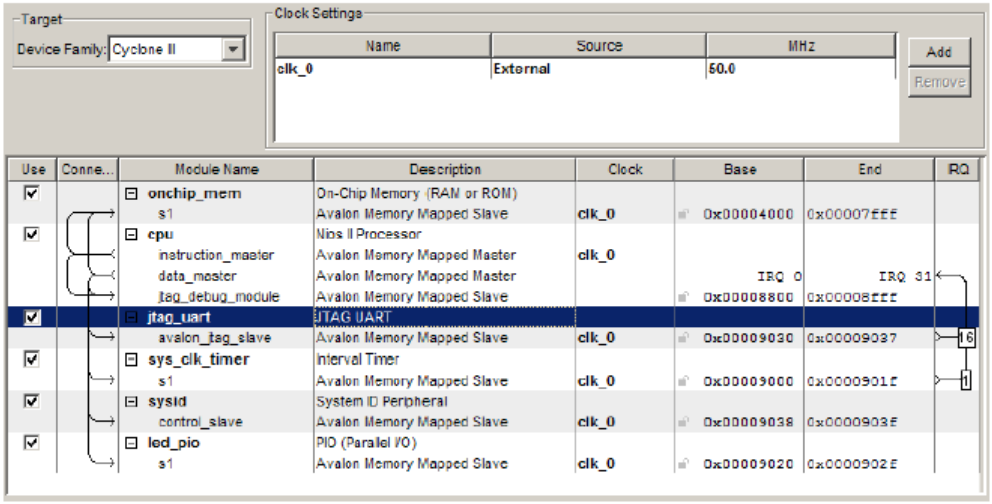
Nios II HAL интерпретирует малые значения IRQ наивысшим приоритетом. Компонент таймера должен иметь наивысший приоритет, чтобы правильно создавать системные такты.

Для назначения нужных базовых адресов и IRQ, выполните следующие пункты:

1. В меню **Система** кликните **Авто назначение базовых адресов**, чтобы позволить SOPC Builder назначить функциональные базовые адреса каждому компоненту системы. **Базовое** и **Конечное** значения в таблице активных компонентов может меняться в зависимости от адресов, переназначаемых SOPC Builder.
2. Кликните на значение **IRQ** для компонента **jtag_uart** и выберите его.
3. Введите 16 и нажмите **Ввод** для назначения нового значения IRQ.

На рисунке 1-13 показано состояние вкладки **Содержимое системы** в SOPC Builder для законченной системы.

Figure 1–13. System Contents Tab with Complete System



Генерирование системы SOPC Builder

Сейчас всё готово для генерирования системы SOPC Builder. Выполните следующие пункты:

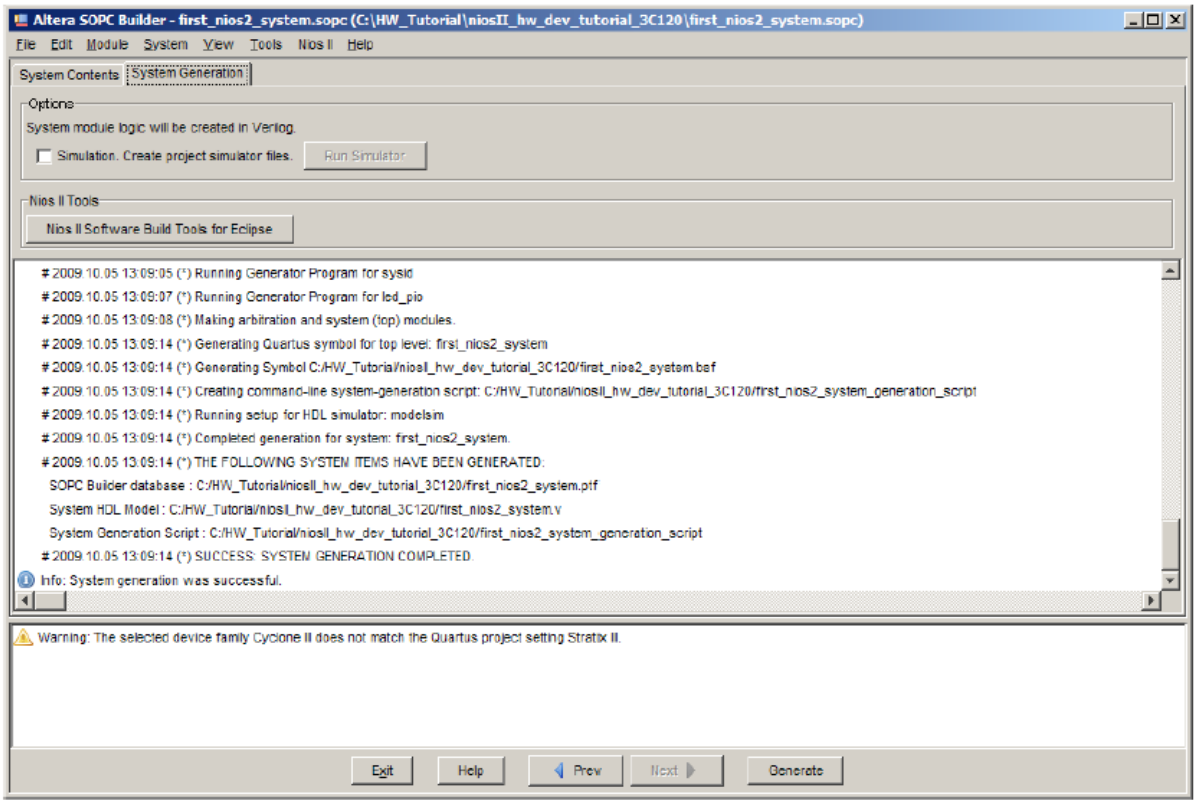
1. Кликните на вкладку **Генерация системы**.
2. Выключите **Симуляция. Создание файлов симулятора проекта.**, чтобы сэкономить время, поскольку этот учебный курс не рассчинан на симуляцию аппаратной части.
3. Кликните **Генерировать**. Появится диалоговое окно **Сохранить изменения**, запрашивающее сохранение вашего проекта.

4. Кликните **Сохранить**. Начнётся процесс генерации системы.

Процесс генерации может занять несколько минут. Когда он закончится, на вкладке **Генерация системы** отобразится сообщение: "Информация: Система сгенерирована успешно."

На рисунке 1-14 показана успешная генерация системы.

Figure 1–14. Successful System Generation



5. Кликните **Выход** для возвращения в программу Quartus II.

Поздравляем! Вы закончили создание системы с процессором Nios II. Теперь вы готовы для интегрирования системы в аппаратный проект Quartus II и для использования инструмента создания программы Nios II под Eclipse для разработки программной части.

За подробной информацией о генерировании системы в SOPC Builder обратитесь к тому 4: *SOPC Builder* в *Настольной книге Quartus II.* Подробнее об аппаратной симуляции системы Nios II, обратитесь к *AN351: Симуляция встроенных процессорных проектов Nios II*.