

Лабораторная работа №5

Проектирование систем на кристалле на основе ПЛИС

Цель работы

Изучение основ построения микропроцессорных систем на ПЛИС. В ходе работы студенты ознакомятся с принципами построения систем на кристалле (СНК) на основе ПЛИС, получат навыки проектирования СНК в САПР Altera Quartus II, выполнят проектирование и верификацию системы с использованием отладочного комплекта Altera DE1Board.

Системы на кристалле

Система на кристалле (SoC, СНК) — это функционально законченная электронная вычислительная система, состоящая из одного или нескольких микропроцессорных модулей, а также системных и периферийных контроллеров, выполненная на одном кристалле. Такая тесная интеграция компонентов системы позволяет достичь высокого быстродействия при построении специализированных ЭВМ. В состав СНК, как правило, включают:

- один или несколько микроконтроллеров, микропроцессоров или ядер цифровой обработки сигналов;
- системные и периферийные шины, арбитры шин и мосты;
- память различного назначения: ПЗУ, ОЗУ, FIFO, Кэш-память;
- контроллеры прямого доступа к памяти;
- контроллер прерываний;
- системы синхронизации и управления сбросом;
- таймеры, счетчики;
- контроллеры стандартных интерфейсов USART, SPI, I2C, Ethernet, USB и пр;
- контроллеры отладки;
- входы и выходы цифро-аналоговых и аналого-цифровых преобразователей.
- регуляторы напряжения и стабилизаторы питания.

Конкретный набор блоков определяется разработчиком в зависимости от назначения системы. Блоки объединяются при помощи внутренних универсальных или специализированных шин, что обеспечивает высокую гибкость при построении системы. В случае, когда СНК реализуются на основе ПЛИС, становится возможным использовать специализированные компоненты даже для мелкосерийных изделий. В связи с этим, такие проекты нередко превосходят аналогичные, выполненные на СБИС, по основным техническим параметрам: быстродействию, надежности, потреблению и пр.

Проектирование СНК на основе ПЛИС разделяют на два этапа: проектирование аппаратного обеспечения и проектирование программного обеспечения. При правильной постановке процесса проектирования, данные этапы могут выполняться параллельно. Проектирование аппаратного обеспечения СНК является наиболее трудоемкой процедурой. Ее можно разделить на следующие этапы:

- Определение номенклатуры входящих в состав СНК устройств;
- Определение типов внутренних и внешних интерфейсных связей;

- Проектирование компонентов системы в соответствии с определенными видами межмодульных связей. Результатом проектирования являются описания на языках описания аппаратных средств, подлежащие последующему синтезу;
- Сборка системы в единый проект, назначение диапазонов адресного пространства шин устройствам системы;
- Задание ограничений проекта: назначения контактов, временных ограничений, начальных состояний распределенной и блочной памяти на целевой ПЛИС.
- Автоматизированные процедуры синтеза, размещение и трассировка аппаратного проекта на ПЛИС;
- Верификация аппаратного обеспечения проекта.

Разработка программного обеспечения опирается на результаты проектирования аппаратного обеспечения, такие как: версии компонентов системы, настоящие параметры «по умолчанию», базовые адреса устройств, алгоритмы инициализации и самоконтроля, алгоритмы обмена информацией с периферийными устройствами и пр. Большинство современных средств проектирования программного обеспечения СНК содержат компиляторы C/C++. Связь программной и аппаратной частей выполняется через заголовочные файлы, включаемые в программный проект. Генерация данных файлов происходит при синтезе аппаратного обеспечения, а использование при компиляции программной части. Таким образом, например, реализована связь через файл `system.h` в САПР Quartus II.

САПР сквозного проектирования систем на кристалле Altera Quartus II

Фирма Altera производит около 30% от общего количества ПЛИС. Помимо самих микросхем с программируемой структурой Altera также предоставляет программные средства проектирования, такие как система сквозного проектирования Quartus II. Эта САПР автоматизирует все этапы проектирования с использованием ПЛИС, от ввода описания устройств до его внутрисхемной верификации.

САПР Altera Quartus II представляет из себя набор модулей для выполнения различных проектных процедур. Все модули объединены в общий маршрут проектирования с помощью центрального программного модуля. Внешний вид его представлен на рисунке 1.

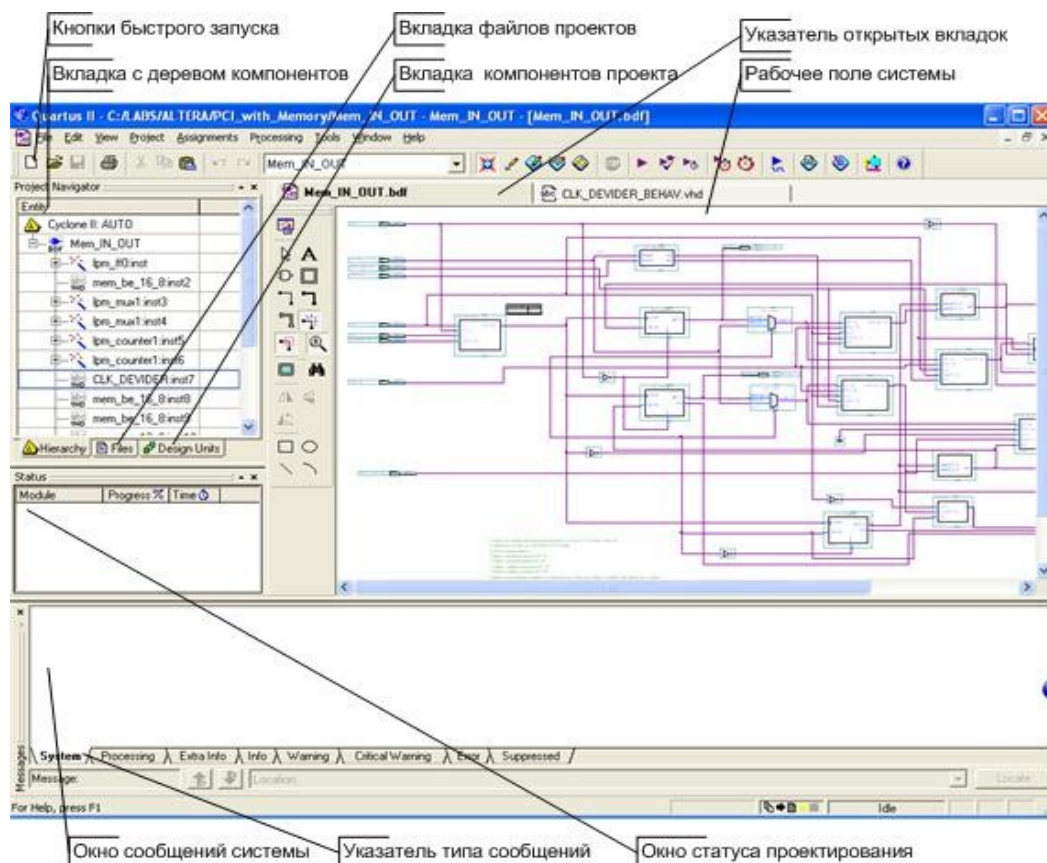


Рисунок 1 — Окно САПР Quartus II.

Реализация систем на кристалле на основе ПЛИС Altera.

В практической части необходимо реализовать в САПР Quartus II систему на кристалле, структура которой представлена на рисунке 2.

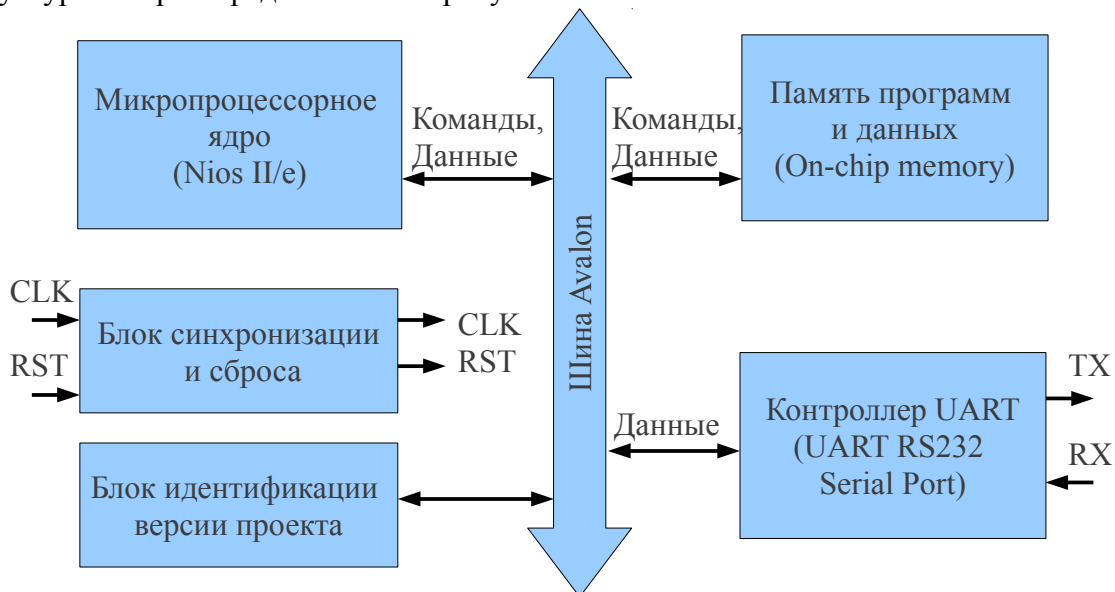


Рисунок 2 — Функциональная схема разрабатываемой системы на кристалле.

Система на кристалле состоит из следующих блоков:

- Микропроцессорное ядро Nios II/е выполняет функции управления системой.
- Внутренняя оперативная память СНК, используемая для хранения программы управления и данных.
- Системная шина Avalon обеспечивает связность всех компонентов системы.
- Блок синхронизации и сброса обеспечивает обработку входных сигналов сброса и синхронизации и распределение их в системе. Внутренний сигнал сброса синхронизирован и имеет необходимую для системы длительность.
- Блок идентификации версии проекта обеспечивает хранение и выдачу уникального идентификатора версии, который используется программой управления при инициализации системы.
- Контроллер UART обеспечивает прием и передачу информации по интерфейсу RS232.

Практическая часть

- 1 Создание проекта в САПР Quartus II.
 - 1.1 Запустить САПР Quartus II (Меню "ПУСК" -> "Программы" -> Altera -> Quartus II 11.0 Web Edition -> Quartus II 11.0 Web Edition)
 - 1.2 В открывшемся окне выбрать «New Project Wizard». Далее нажать кнопку «Next».
 - 1.3 Указать путь к директории проекта: c:\user\sopc01. Имя проекта: socp01. Далее нажать кнопку «Next» 2 раза.
 - 1.4 Выбрать целевую ПЛИС. Family: Cyclone II. Package: FBGA. Pin count: 484. Speed grade: 7. В таблице Available devices указать: EP2C20F484C7.
 - 1.5 Далее нажать кнопку «Next» и «Finish». Результат создания проекта показан на рисунке 3.

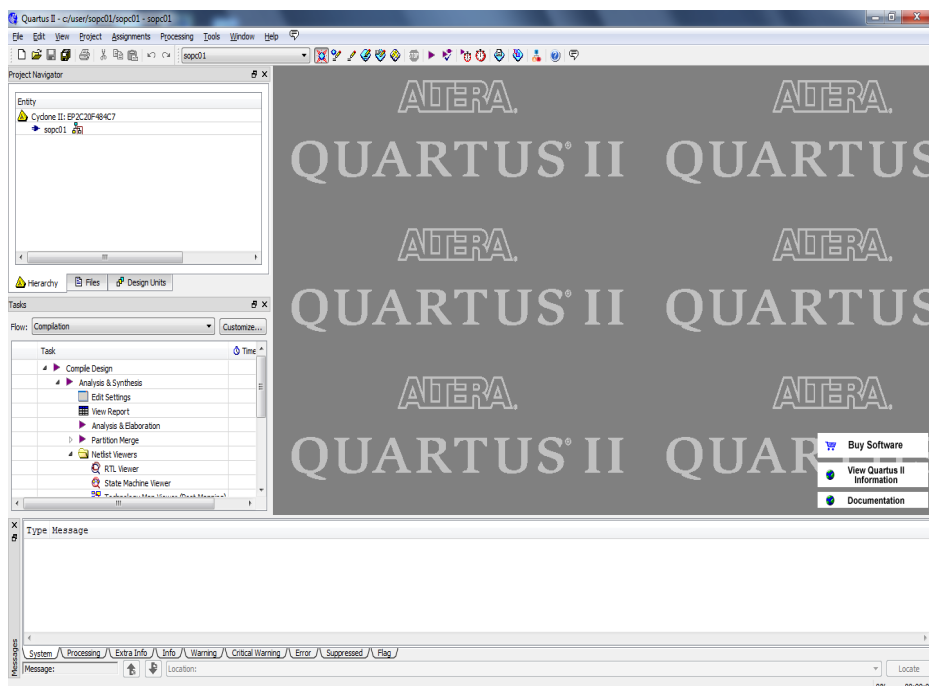


Рисунок 3 — Новый проект в САПР Quartus II 11.0 Web Edition.

- 2 Создать новый модуль системы на кристалле QSYS.

- 2.1 В меню «File» выбрать пункт «New».
- 2.2 В разделе «Design Files» выбрать пункт «Qsys System File». Нажать кнопку «Ok». В результате будет создан новый модуль ЧК, содержащий компонент «Clock and Reset Bridge» (Рисунок 4).

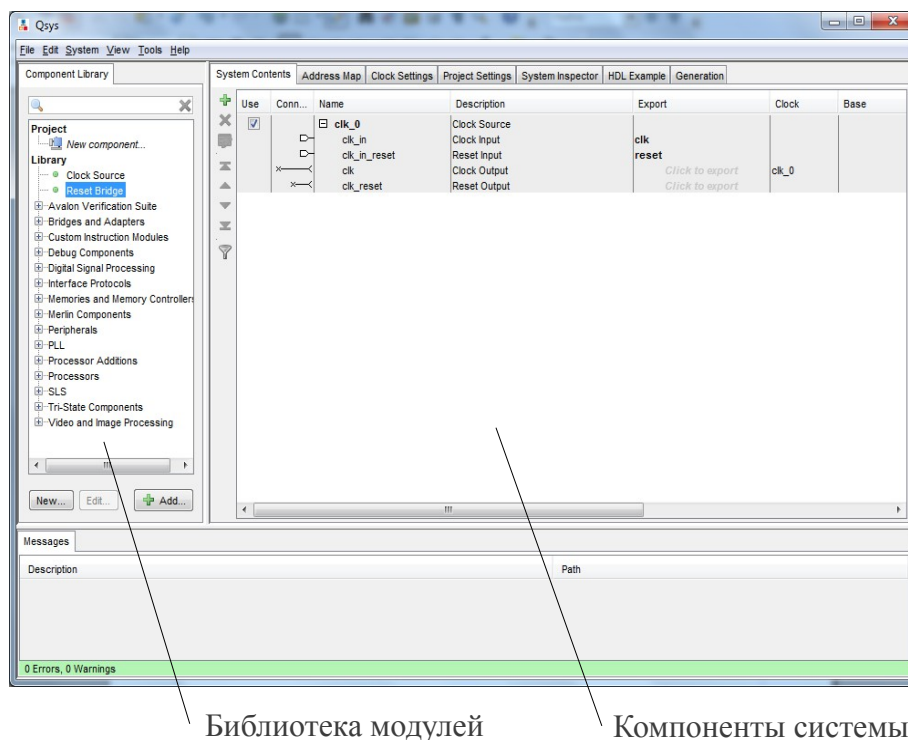


Рисунок 4 — Модуль Qsys.

- 2.3 Установить частоту внешнего сигнала синхронизации (дважды нажать на компонент clk_0, параметра Clock frequency) 50 000 000 Гц.
- 2.4 Сохранить модуль Qsys в файл c:\user\sopc01\nios.qsys.
- 2.5 Добавить в проект модуль синтезируемого микропроцессорного ядра Nios2 (Embedded Processors → Nios II Processors).
- 2.6 В открывшемся диалоге настройки компонента выбрать тип ядра: Nios II/e (см. рисунок 5).

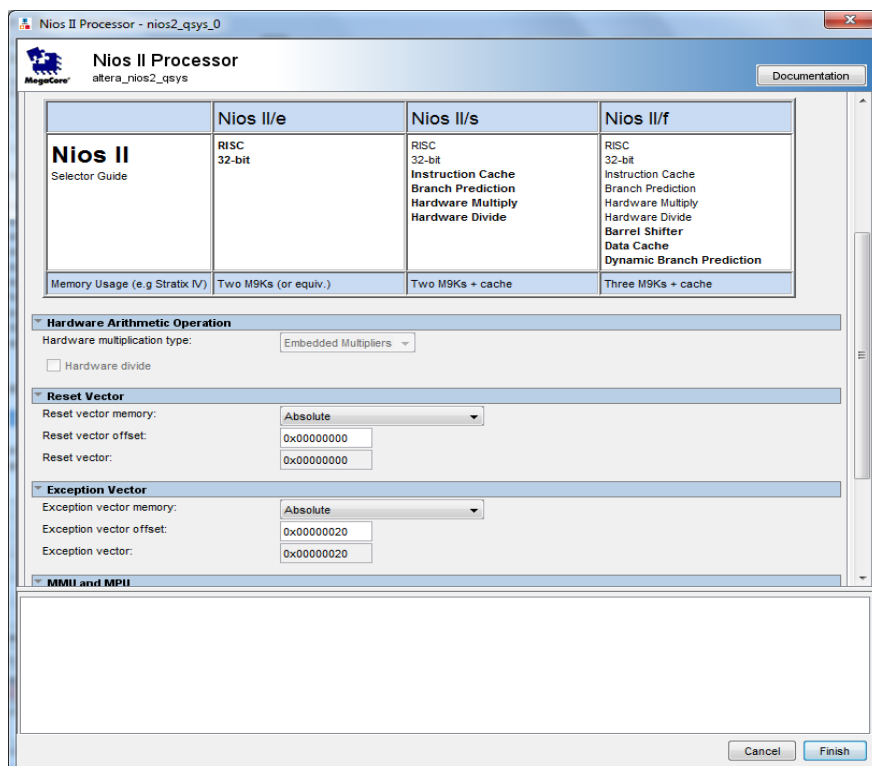


Рисунок 5 - Настройка параметров компонента Nios

- 2.7 Добавить в проект модуль ОЗУ программ и данных: Memory and Memories Controller → On-Chip → On-Chip Memory (RAM and ROM). Выбрать в открывшемся окне размер памяти: 8196 байт. Размер слова: 32. Остальные параметры оставить без изменений. Нажать кнопку «Finish».
- 2.8 Добавить компонент Avalon System ID аппаратного идентификатора версии: Peripherals → Debug and Performance → System ID Peripheral. Настройка компонента не требуется (можно сразу задать System ID - см. пункт 12).
- 2.9 Добавить компонент Avalon UART: Interface protocols → Serial → UART (RS232 Serial Port). При настройке использовать параметры: Parity: None; Data bits: 8; Stop bits: 1; Baud rate: 115200. Остальные параметры оставить без изменения.
- 2.10 Создать сеть синхронизации и сброса системы: На вкладке System Contents соединить выход блока синхронизации (Clock Output) и сброса (Reset Output) с входами clk (clk1) и reset (reset1) блоков Nios II/e, Onchip Memory, System ID peripheral и UART.
- 2.11 Подключить все блоки к системной шине Avalon: выходы Data master и Instruction Master процессора NIOSII/e подключить к: входу s1 блока Onchip Memory; к входу control slave блока System ID Peripheral; к входу s1 блока UART.
- 2.12 Экспортировать сигналы TX и RX во внешние порты. Для этого во вкладке System Contents выбрать компонент UART и сигнал external connections. В столбце Export указать название сигнала: uart0.
- 2.13 Соединить выход IRQ блока UART с входом IRQ процессора (выполняется в столбце IRQ).
- 2.14 Выполнить настройку таблицы прерываний процессора Nios II/e. Для этого выбрать компонент nios2_qsys_0 и перейти к редактированию его свойств (двойное нажатие на компонент). В открывшемся диалоге настройки компонента в поле Reset Vector выбрать параметр Reset Vector Memory: onchip_memory2_0.s1.
- 2.15 В поле Exception Vector выбрать параметр Exception Vector Memory: onchip_memory2_0.s1. Нажать кнопку «Finish».

2.16 Назначить базовые адреса устройств: в меню System → Assign Base Addresses. После данного действия окно Qsys должно иметь вид, показанный на рисунке 6.

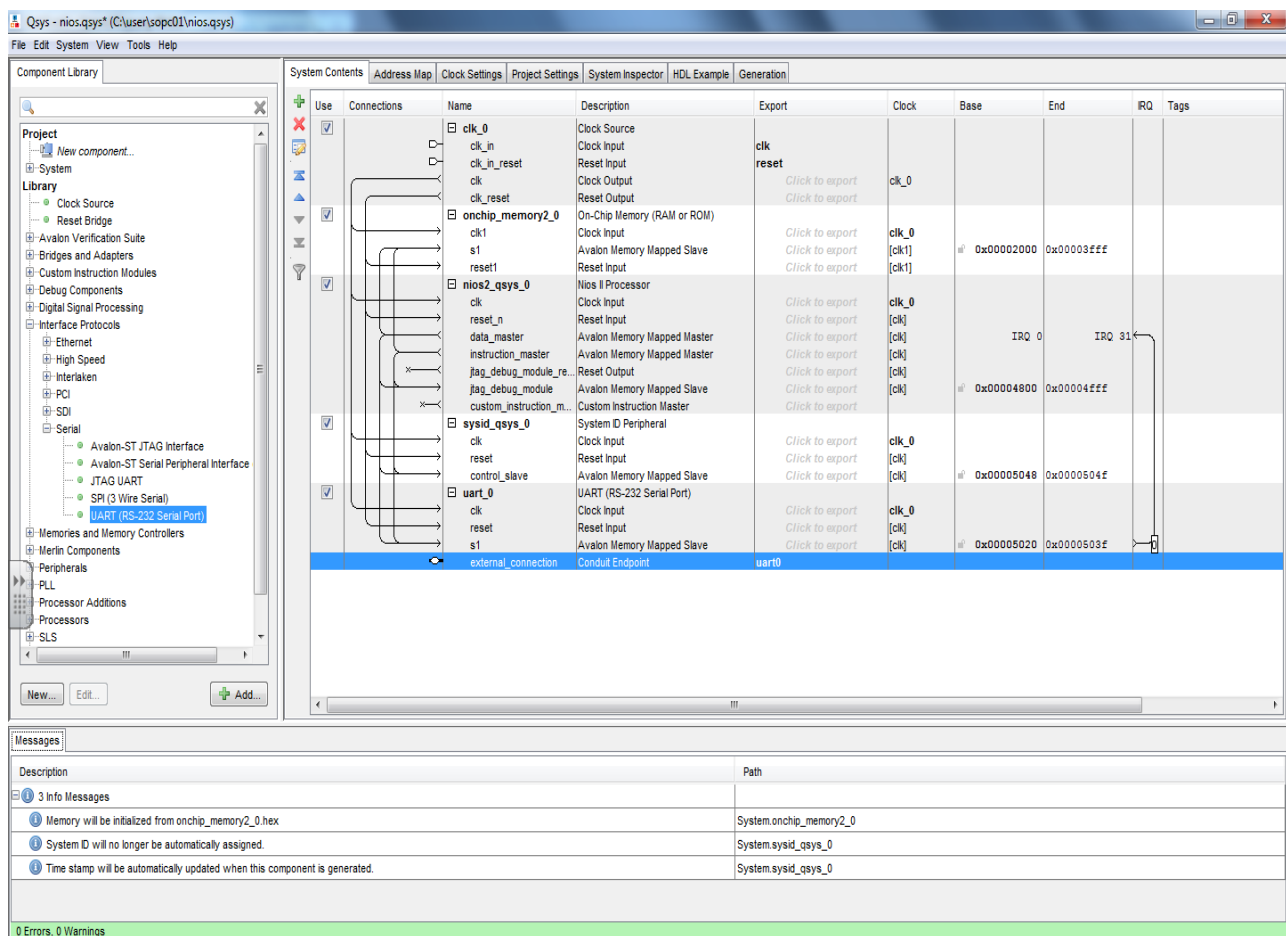


Рисунок 6 — Окно модуля Qsys после назначения базовых адресов.

- 2.17 Сохранить изменения в модуле Qsys.
- 3 Добавить модуль c:\user\sopc01\nios.qsys в проект socp01. Для этого в меню Project САПР Quartus II выбрать пункт Add/Remove Files in Project. В открывшемся окне указать в поле File name: c:\user\sopc01\nios.qsys (см. рисунок 7).

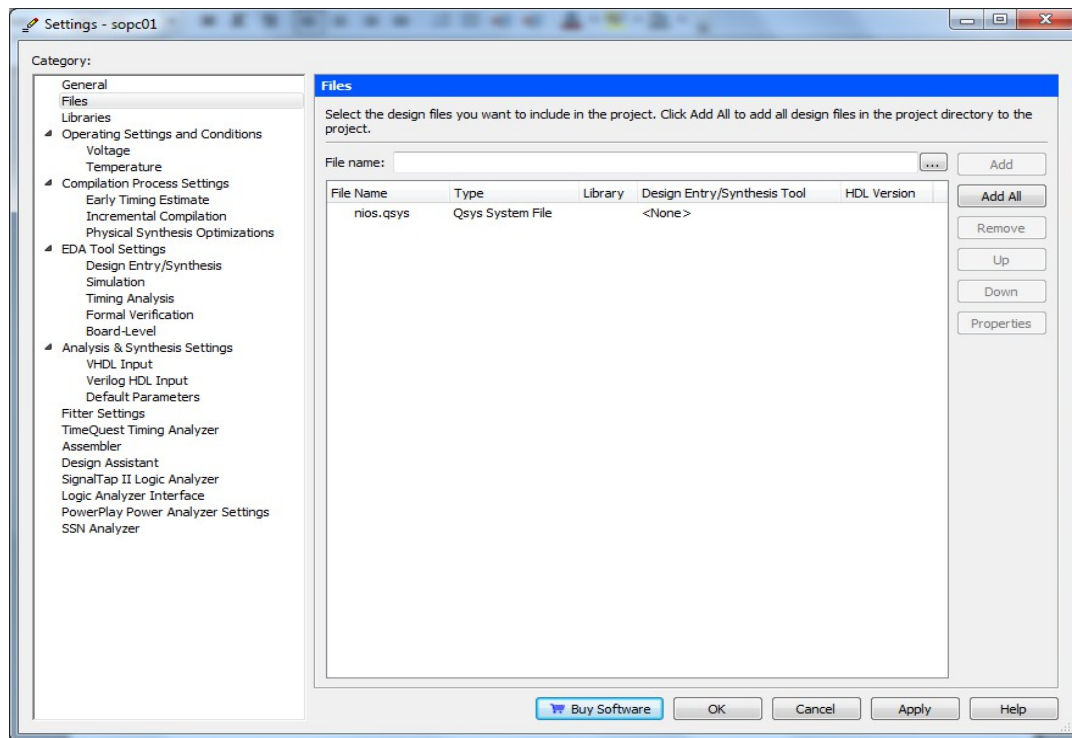


Рисунок 7 — Добавление модуля nios.qsys в проект.

- 4 Назначить модуль nios.qsys в качестве модуля верхнего уровня. Для этого в окне Project Navigator (в левом верхнем углу) выбрать вкладку File. Выбрать модуль nios.qsys и нажать Ctrl-Shift-J (или нажать правой кнопкой мыши и выбрать Set as Top-level Entity).
- 5 Выполнить синтез проекта. Для этого в меню Processing выбрать Start compilation.
- 6 Назначить портам проекта контакты микросхемы.
 - 6.1 В окне Tasks (в левом нижнем углу) выбрать пункт: Compile Design → Analysis & Synthesis → I/O Assignment Analysis → Pin planner (см. рисунок 8).

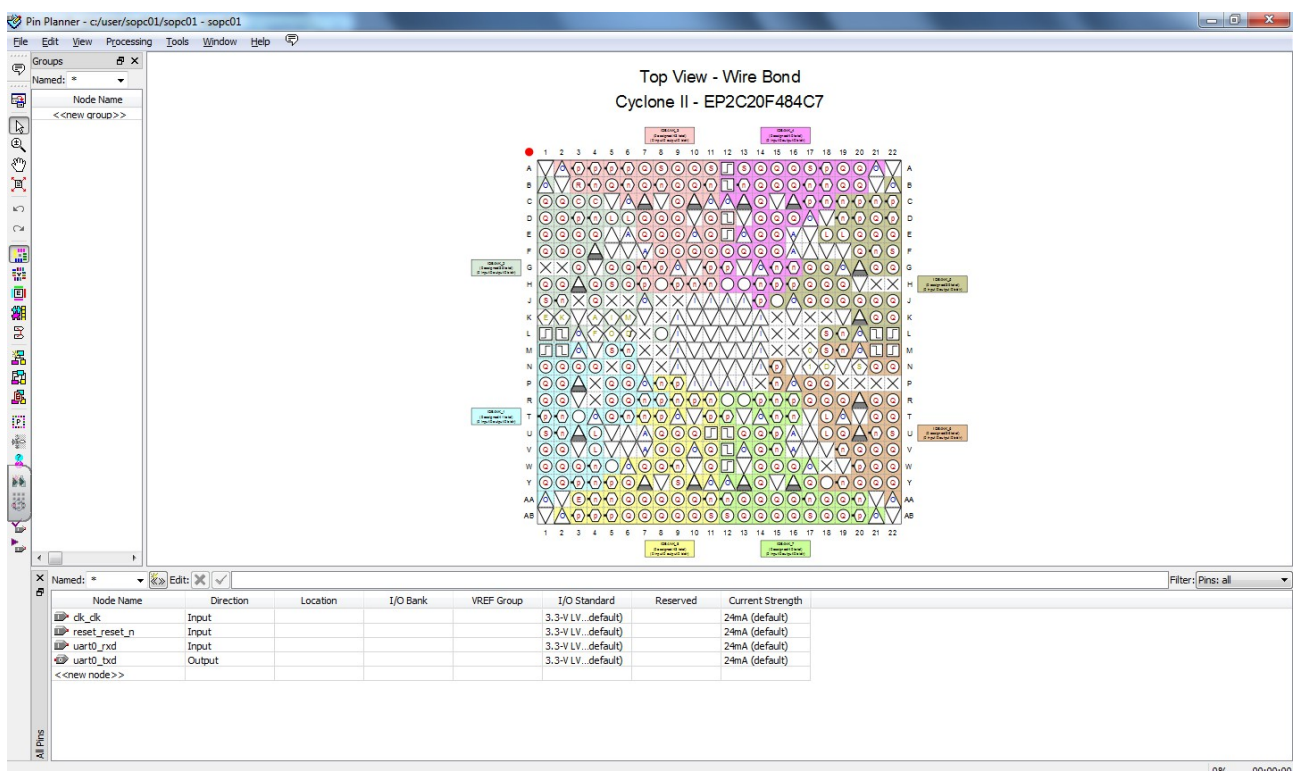


Рисунок 8 — Модуль Pin Planner.

6.2 Назначить контакты в соответствии с таблицей 1.

Таблица 1. Назначение контактов микросхемы портам проекта

Сигнал	Контакт
clk	L1
reset	R22
uart0_rxd	F14
uart0_txd	G12

- 7 Выполнить синтез проекта. Для этого в меню Processing выбрать Start compilation. В случае успешного завершения на экране появится окно с соответствующим сообщением.
- 8 Создать программный проект Nios2.
 - 8.1 Запустить Nios II Software Build Tools for Eclipse. Для этого в меню Tools в САПР Quartus II выбрать Nios II Software Build Tools for Eclipse (далее, просто SDK).
 - 8.2 Указать путь для сохранения проекта SDK: c:\user\sopc01\.
 - 8.3 Создать шаблон проекта “Hello_world”. Для этого в меню File выбрать пункт New. Далее выбрать пункт Nios II Application and BSP Template.
 - 8.4 В открывшемся диалоговом окне настройки проекта указать:
SOPC Information File Name: C:\user\sopc01\nios.sopcinfo.
Project name: hello.
Project Template: Hello World Small.
 - 8.5 Нажать кнопку Next и Finish. В результате должны быть созданы два проекта: проект hello_bsp (Board support package) содержащий библиотеки функций, настроенные на данный проект; шаблон проекта hello, содержащий примитивный код теста UART.
 - 8.6 Добавить в код файла hello_world_small.c код эхо-программы приема-передачи по интерфейсу RS232:

```
#include "sys/alt_stdio.h"

int main()
{
    char ch;
    alt_putstr("Hello from System on Chip\n");
    alt_putstr("Send any character\n");

    /* Event loop never exits. */
    while (1) {
        ch=alt_getchar();
        alt_putchar(ch);
    }

    return 0;
}
```

- 8.7 Создать образ ОС HAL с драйверами устройств, используемых в аппаратном проекте. Для этого в меню Nios II → BSP Editor. В открывшемся окне выбрать меню File → Open и указать путь к файлу settings.bsp, а именно C:\user\sopc01\Software\hello_bsp\. Нажать на кнопку Generate в нижнем правом углу окна Nios II BSP Editor.
- 8.8 Выполнить сборку программного проекта: Меню Project → Build All.
- 9 Выполнить прошивку проекта в ПЛИС.
 - 9.1 В САПР Quartus II в меню Tasks (левый нижний угол) Program Device (см. рисунок 9).

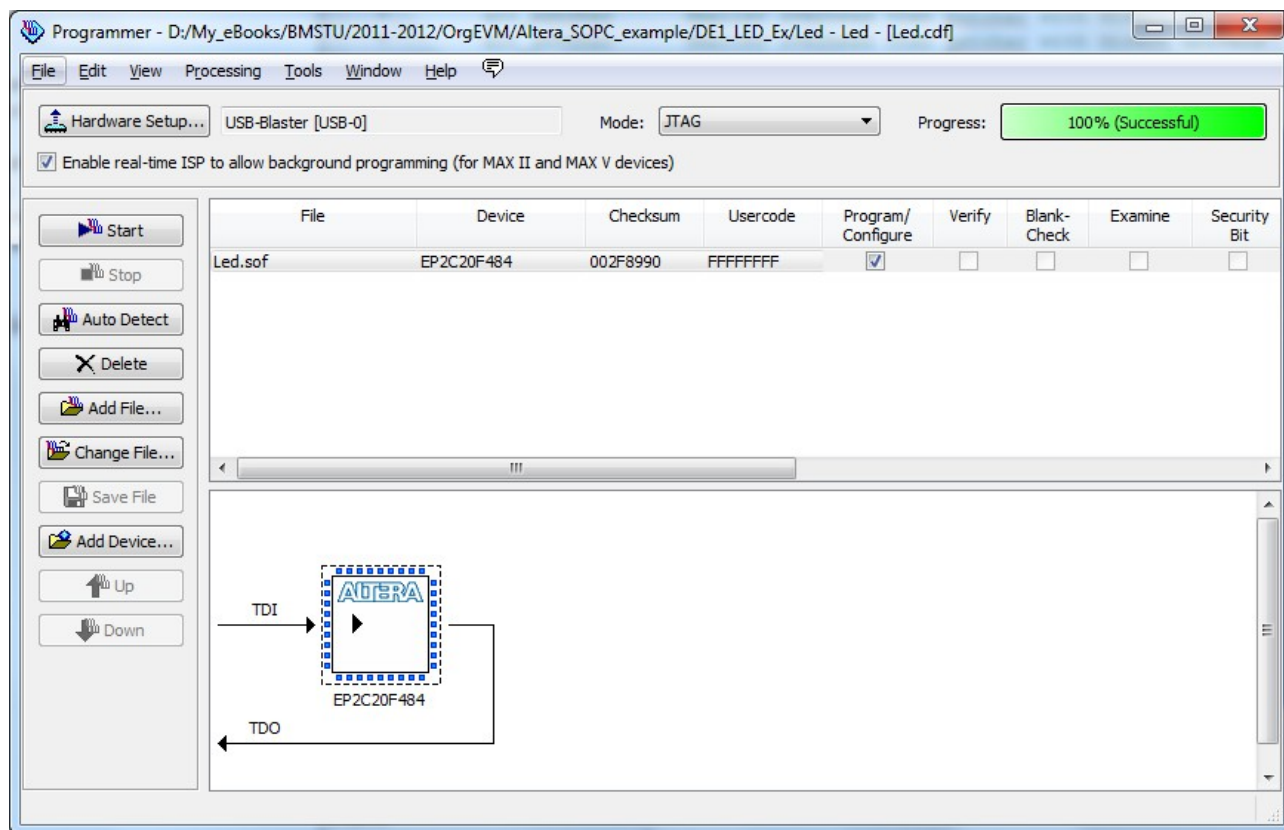


Рисунок 9 — Окно модуля программирования ПЛИС

- 9.2 Подключить к ПК отладочную плату с ПЛИС EPC2C20.
- 9.3 Произвести автоматический поиск устройства программирования: нажать кнопку Hardware Setup.
- 9.4 В случае успешного поиска программатора указать файл CHK socp01.sof.
- 9.5 Выполнить программирование, нажав кнопку Start.
- 10 Выполнить загрузку тестового программного проекта hello в ПЛИС с использованием Nios II Software Build Tools for Eclipse.
 - 10.1 Подключить плату Altera Cyclon II FPGA Starter Board к USB разъему компьютера.
 - 10.2 В Nios II Software Build Tools for Eclipse выбрать пункт Run->Debug Configuration.
 - 10.3 Создать новую Debug – сессию типа “Nios II Connection Hardware”.
 - 10.4 В поле Project Name выбрать проект hello.
 - 10.5 Убедиться в активности соединения с USB-Blaster на вкладке Target Connection (В таблице Processors должна быть указана строка с процессором Nios2).
 - 10.6 На вкладке “Target Connections” отметить пункты:

- Ignore mismatched system ID
- Ignore mismatched system timestamp
- Reset the selected target system

10.7 Нажать кнопку Debug.

- 11 Выполнить верификацию проекта с использованием программы терминала.
- 12 Изменить параметр System ID на 32-х разрядный код, состоящий из номера группы и варианта. Например «5215» означает группу ИУ7-52, вариант 15. Номер задается в десятичном формате (не в hex).
- 13 Доработать код программного проекта: добавить строки, передающие по UART значение SystemID в виде четырех байт символов в ASCII формате.

Для этого следует использовать файлы описаний, находящиеся в BSP проекте:

```
#include "system.h"  
#include "altera_avalon_sysid_qsys.h"  
#include "altera_avalon_sysid_qsys_regs.h"
```

Для получения System ID номера использовать функцию IORD_ALTERA_AVALON_SYSID_QSYS_ID(SYSID_QSYS_0_BASE), где SYSID_QSYS_0_BASE — базовый адрес блока SystemID (записывается в файл system.h автоматически при сборке проекта).

- 14 Выполнить верификацию проекта с использованием программы терминала. Код программной части проекта занести в отчет.