|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ **Информатика и системы управления**

КАФЕДРА **«Программное обеспечение ЭВМ и информационные технологии»**

**Отчет**

|  |  |
| --- | --- |
| **по практикуму №** | 2 |

**Название:**

Обработка и визуализация графов в вычислительном

комплексе Тераграф

**Дисциплина:** Архитектура ЭВМ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Студент | ИУ7И-54Б |  |  | Динь Вьет Ань |
|  | (Группа) |  | (Подпись, дата) | (И.О. Фамилия) |
|  |  |  |  |  |
| Преподаватель |  |  |  | А.Ю.Попов |
|  |  |  | (Подпись, дата) | (И.О. Фамилия) |

**Цель работы:** изучение основ построения микропроцессорных систем на ПЛИС. В ходе работы необходимо ознакомиться с принципами построения систем на кристалле (СНК) на основе ПЛИС, получить навыки проектирования СНК в САПР Altera Quartus II, выполнить проектирование и верификацию системы с использованием отладочного комплекта Altera DE1Board.

1. **Функциональная схема разрабатываемой системы на кристалле.**

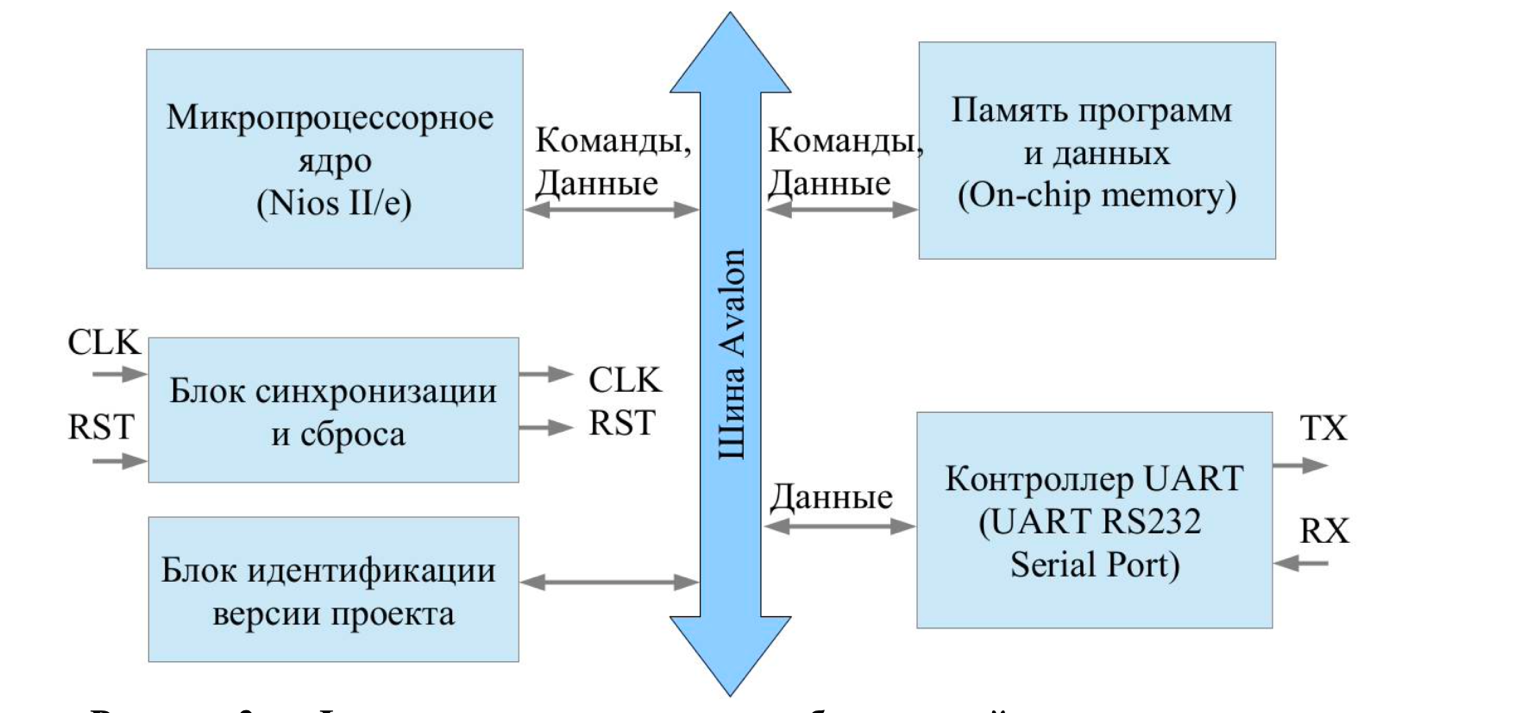


Рисунок 1 Функциональная схема разрабатываемой системы на кристалле

Система на кристалле состоит из следующих блоков:

* Микропроцессорное ядро Nios II/e 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.

1. **Скриншоты маршрута проектирования**
   1. **Модуль в QSYS**
2. Был создан новый модуль Qsys.
3. Установлена частота внешнего сигнала синхронизации 50 000 000 Гц.
4. Добавлен в проект модуль синхронизируемого микропроцессорного ядра Nios2.
5. Добавлен в проект модуль ОЗУ программ и данных.
6. Добавлены компоненты Avalon System ID, Avalon UART.
7. Создана сеть синхронизации и сбоса системы.
8. Сигналы TX и RX экспортированы во внешние порты.
9. Назначены базовые адреса устройств.

Итог выполненных действий показан на рисунке 2. На рисунке 3 показана таблица распределения адресов.

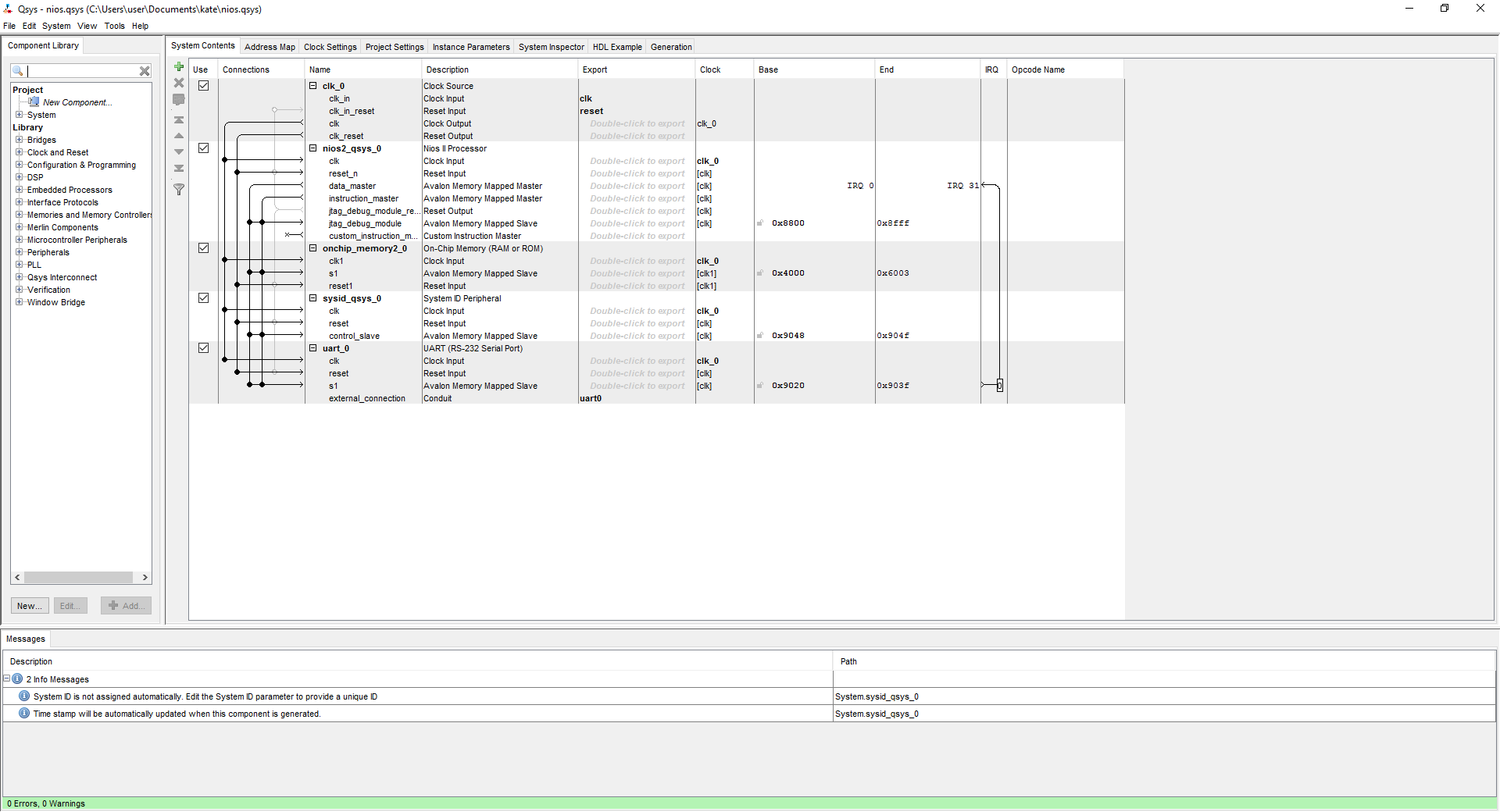


Рисунок 2 Модуль Qsys

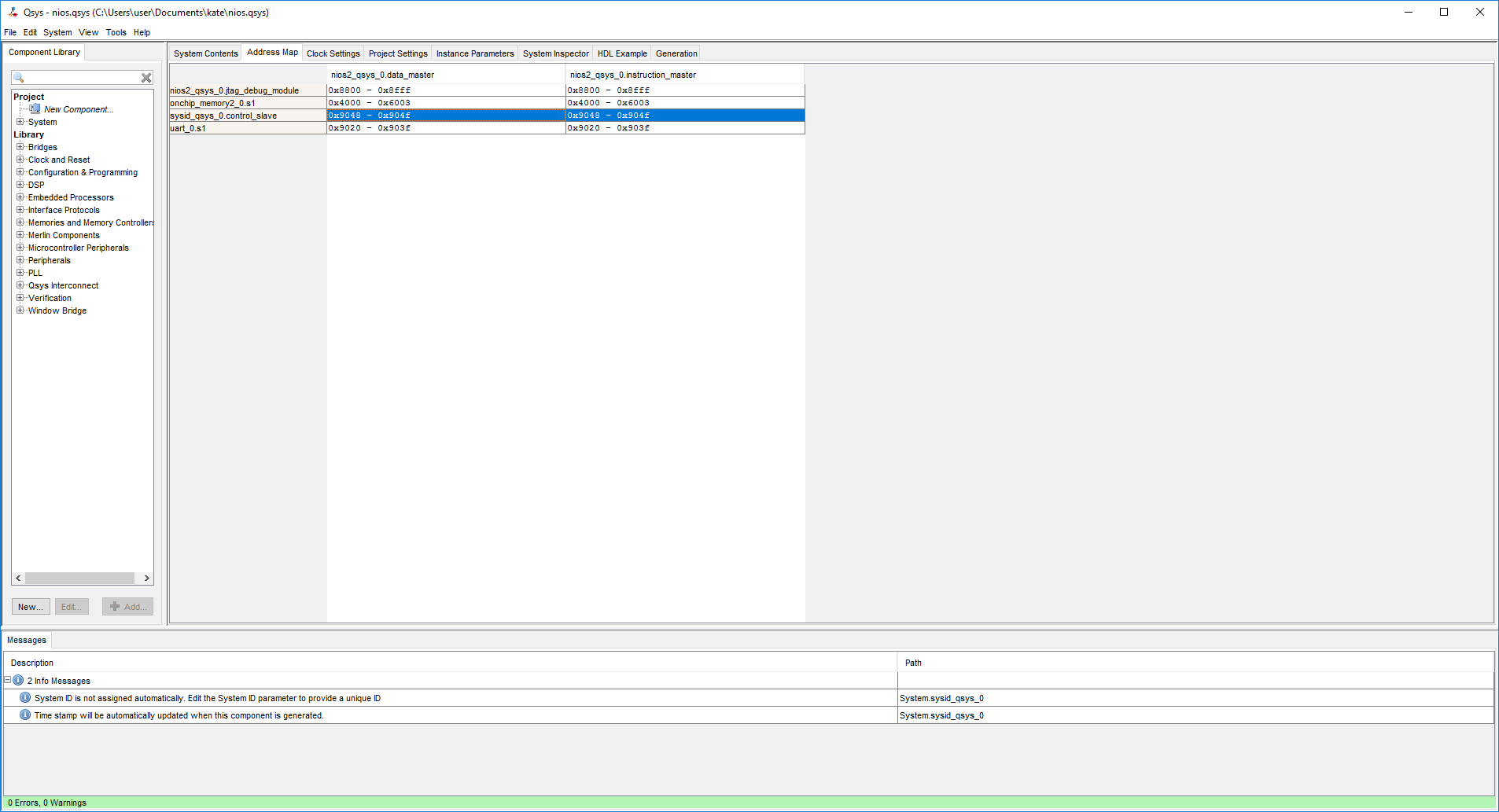


Рисунок 3 Таблица распределения адресов

* 1. **Назначение портам проекта контактов микросхемы**

Был выполнен синтез проекта. Результат показан на рисунке 4.

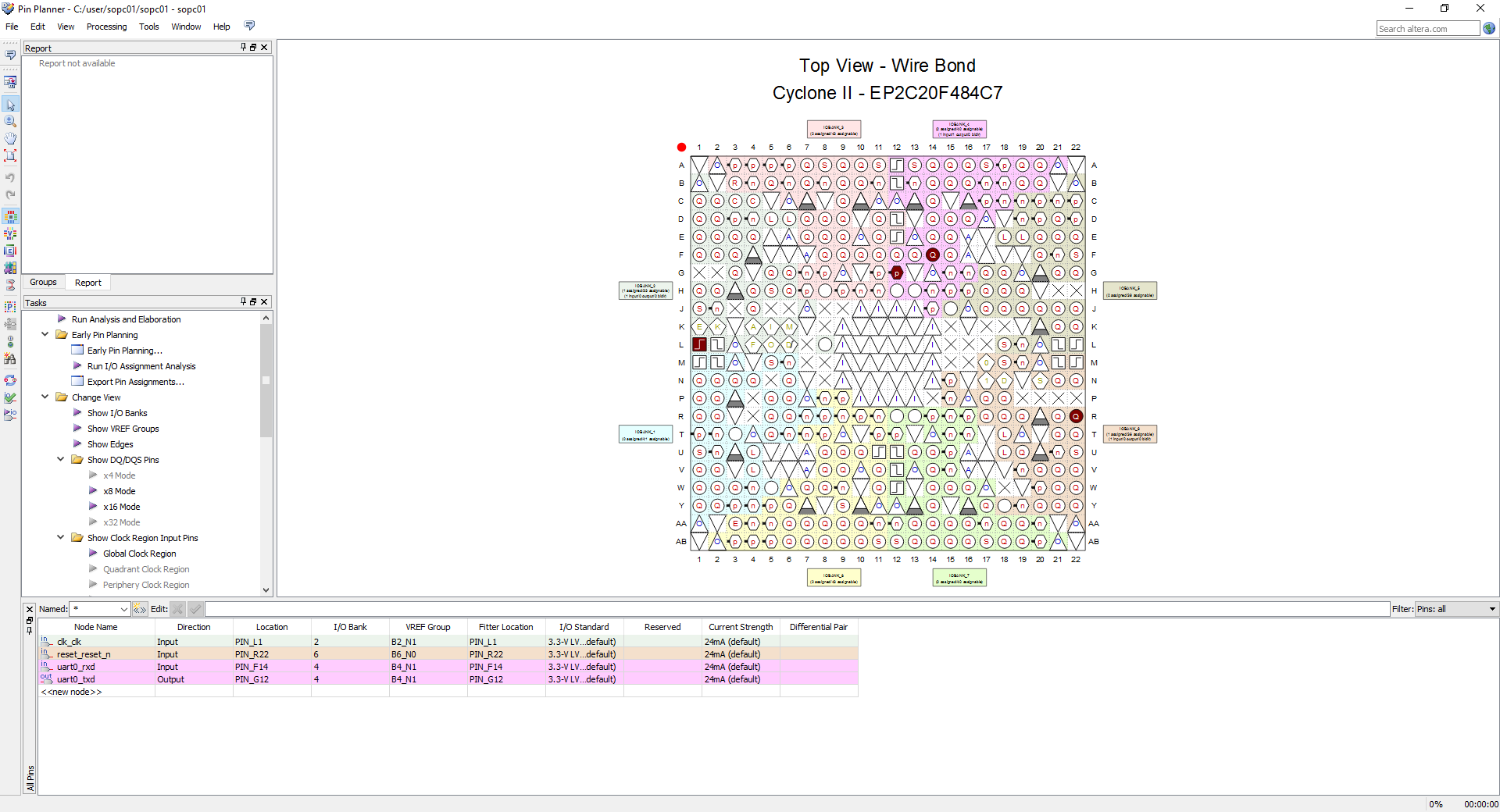


Рисунок 4 Pin Planner

Далее были назначены портам проекта контакты микросхемы в соответствии с таблицей 1, после чего был выполнен повторный синтез проекта.

Таблица 1 Назначение контактов микросхемы портам проекта

|  |  |
| --- | --- |
| **Сигнал** | **Контакт** |
| clk | L1 |
| reset | R22 |
| uart0\_rxd | F14 |
| uart0\_txd | G12 |

* 1. **Создание проекта Nios2**

В файл hello\_world\_small.c был добавлен код эхо-программы приема-передачи по интерфейсу RS232, представленный на листинге 1. Также был создан образ ОС HAL с драйверами устройств, используемых в аппаратном проекте.

*Листинг 1. 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;**

**}**

После успешной сборки и выполнения код программы был доработан: были добавлены строки, передающие по UART значение SystemID в виде четырех байт символов в ASCII формате.. Результат доработки представлен на листинге 2.

*Листинг 2. Код эхо-программы приема-передачи по интерфейсу RS232*

**#include "sys/alt\_stdio.h"**

**#include "system.h"**

**#include "altera\_avalon\_sysid\_qsys.h"**

**#include "altera\_avalon\_sysid\_qsys\_regs.h"**

**int main()**

**{**

**char ch;**

**alt\_putstr("Hello from System on Chip\n");**

**alt\_putstr("Send any character\n");**

**int id = IORD\_ALTERA\_AVALON\_SYSID\_QSYS\_ID(SYSID\_QSYS\_0\_BASE);**

**char a[6];**

**int i = 1;**

**while (id)**

**{**

**a[4 - i] = '0' + id % 10;**

**id /= 10;**

**i++;**

**}**

**a[4] = '\n';**

**a[5] = '\0';**

**for (int i = 0; i < 5; i++)**

**alt\_putchar(a[i]);**

**/\* Event loop never exits. \*/**

**while (1) {**

**ch=alt\_getchar();**

**alt\_putchar(ch);**

**}**

**return 0;**

**}**

Результат выполнения программы показан на рисунке 5.

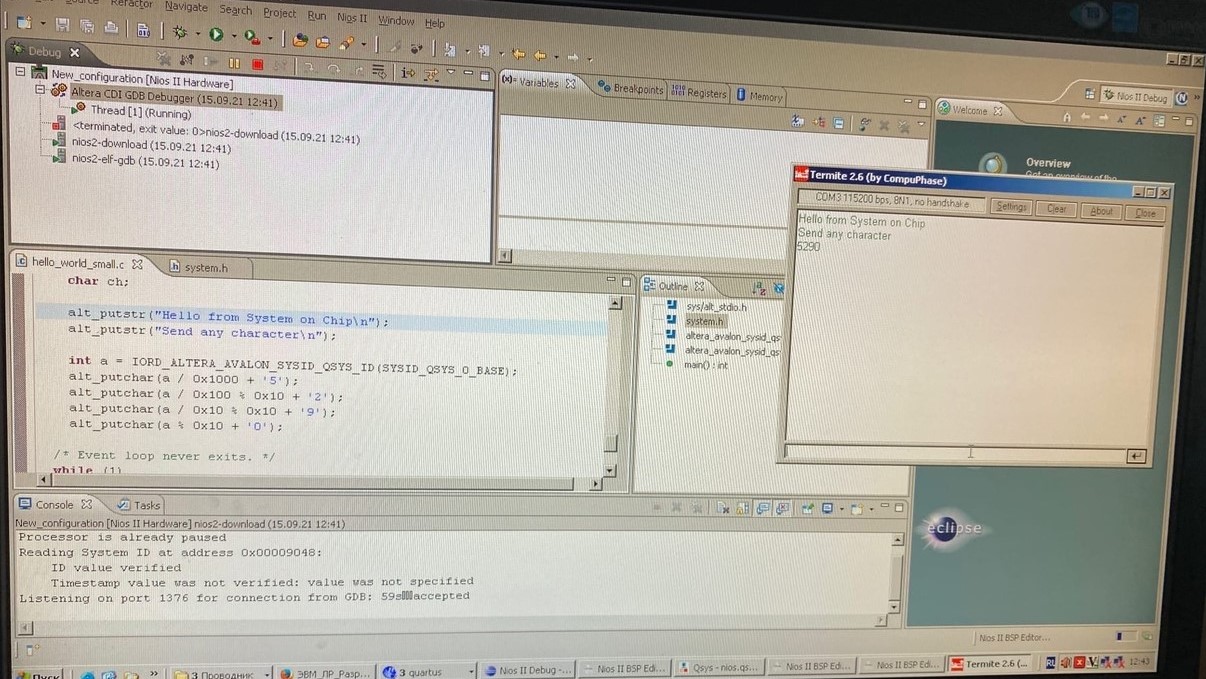


Рисунок 5 Результат выполнения доработанного кода

**Вывод**

В ходе данной лабораторной работы были изучены основы построения микропроцессорных систем на ПЛИС, получены навыки проектирования СНК в САПР Altera Quartus II, также были выполнены проектирование и верификация системы с использованием отладочного комплекта Altera DE1Board.

Поставленная **цель** достигнута.