Санкт-Петербургский политехнический университет Петра Великого
Институт компьютерных наук и кибербезопасности
Высшая школа компьютерных технологий и информационных систем

Отчёт по лабораторной работе № 11

Дисциплина: Автоматизация проектирования дискретных устройств (на английском языке).

Выполнил студент гр. 5130901/10101 _____ Д.Л. Симоновский
(подпись)

Руководитель _____ А.П. Антонов
(подпись)

“21” апреля 2024 г.

Санкт-Петербург
2024

Оглавление

1. Список иллюстраций: 2
2. Задание: 3
   2.1. Цель задания: 3
   2.2. Структура проекта: 3
3. Ход работы: 3
4. Вывод: 13

1. Список иллюстраций:

Рис. 2.1. Структура проекта. 3
Рис. 3.1. Создание проекта. 4
Рис. 3.2. Модуль Clock_Source. 4
Рис. 3.3. Настройки On-Chip Memory. 4
Рис. 3.4. Добавление процессора. 5
Рис. 3.5. Отключение JTAG Debug. 5
Рис. 3.6. Подключения модулей в проекте. 5
Рис. 3.7. Настройка параметров в модуле nios2_PD. 5
Рис. 3.8. Модуль I/O для светодиодов. 6
Рис. 3.9. Настройка подключений светодиодов к процессору. 6
Рис. 3.10. Модуль I/O для переключателей. 7
Рис. 3.11. Подключение SW к процессору. 7
Рис. 3.12. Окно Address Map. 7
Рис. 3.13. Предустановки системы. 8
Рис. 3.14. Окно Messages. 8
Рис. 3.15. RTL Viewer. 9
Рис. 3.16. Входы-выходы в Pin Planner. 9
Рис. 3.17. Создание проекта в Nios II. 9
Рис. 3.18. Создание source файла. 10
Рис. 3.19. Результат компиляции. 10
Рис. 3.20. Измеренные настройки генерации. 11
Рис. 3.21. Повторная компиляцию. 11
Рис. 3.22. Изменение настроек вводов-выводов. 12
Рис. 3.23. Pin Planner. 12

2. Задание:

2.1. Цель задания:

Введение в реализацию "системы внутри кристалла" представляет собой проект, основанный на использовании процессора NIOSII, включающий в себя следующие этапы:

- Начало работы с проектом в среде Quartus Prime (QP)
- Создание аппаратной части проекта с использованием инструмента Platform Designer (PD)
- Разработка программной части проекта в рамках среды NIOSII IDE
- Проверка функционирования проекта на платформе.

2.2. Структура проекта:

Визуализация чисел от 0 до 255 в двоичном коде происходит на светодиодах LED1 до LED8 через процессор NIOSII. Этот процесс контролируется данными, поступающими с переключателей SW.:

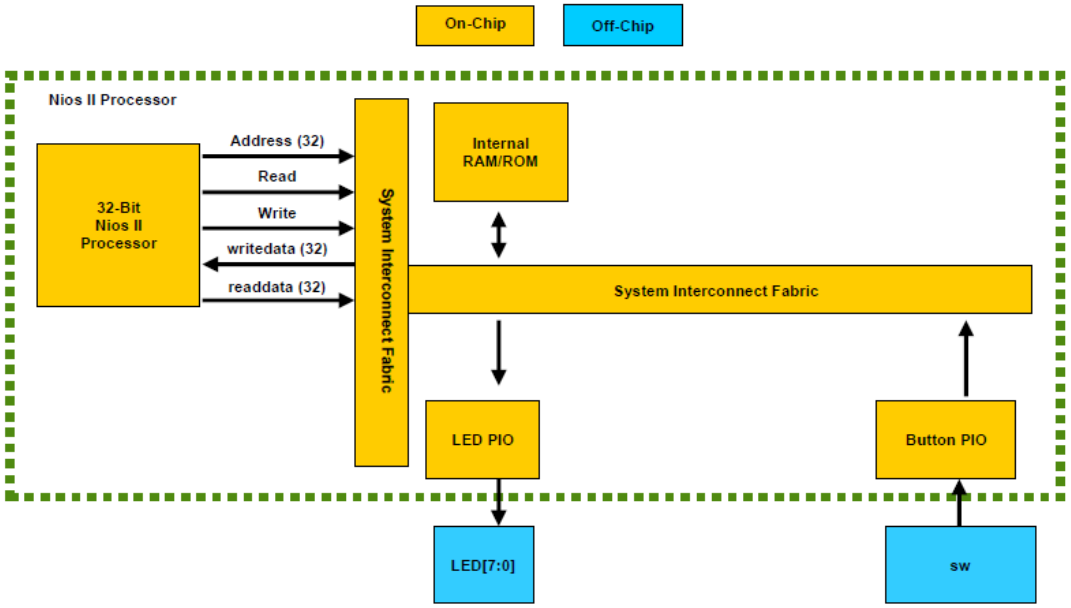


Рис. 2.1. Структура проекта.

Конфигурация устройства включает в себя главный модуль и три вспомогательных: два модуля подчиненных my_slave и один модуль my_Dslave (стандартный подчиненный). Главный модуль получает данные через Conduit и через 8-битный интерфейс осуществляет адресное взаимодействие с одним из подчиненных, настраивая их соответственно либо записывая что-то в них. У каждого из подчиненных есть собственный Conduit, который обеспечивает возможность просмотра того, что было записано в него из главного модуля.

3. Ход работы:

Выполним создание проекта со следующими настройками:

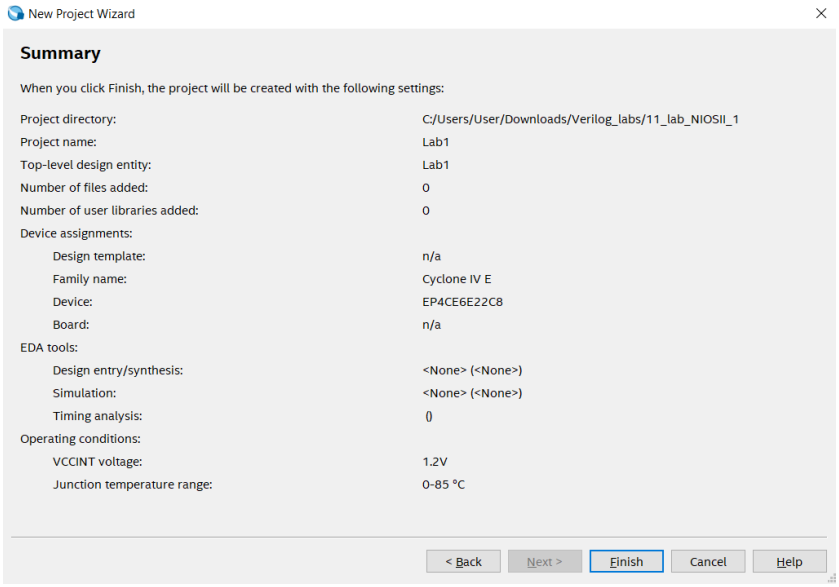


Рис. 3.1. Создание проекта.

Откроем Platform Designer и выполним настройку созданного модуля clk:

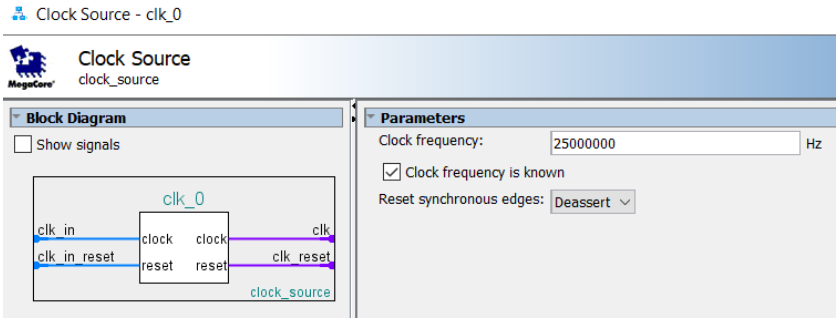


Рис. 3.2. Модуль Clock_Source.

Установили синхронизацию для reset и частоту, равную частоте устройства.

Далее добавим модуль onchip_mem, который будет хранить программу процессора. Тип памяти RAM, размер 16 Кб, остальные настройки приведены ниже:

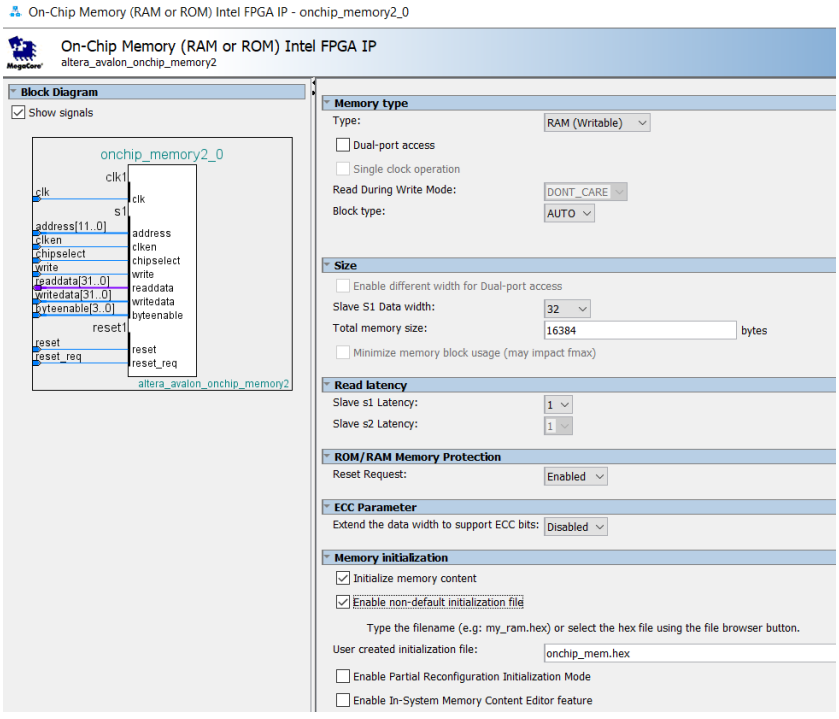


Рис. 3.3. Настройки On-Chip Memory.

Добавим сам процессор:

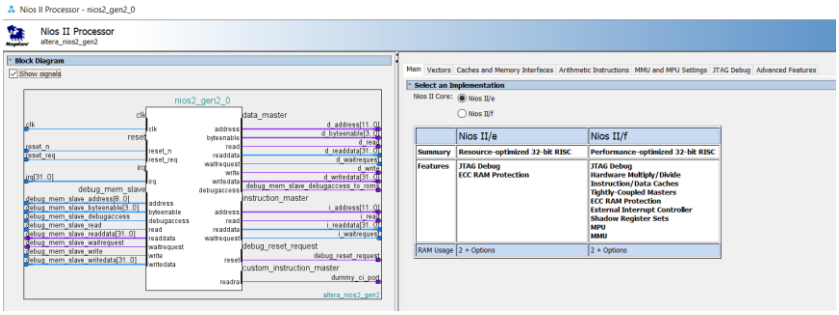


Рис. 3.4. Добавление процессора.

Выключим JTAG debug т. к. на данный момент мы не будем это использовать:

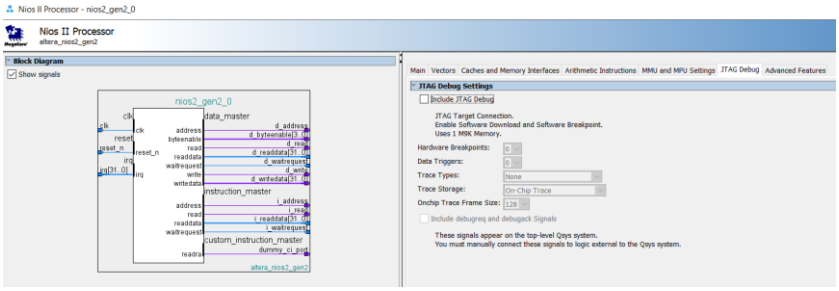


Рис. 3.5. Отключение JTAG Debug.

Выполним подключения тактового сигнала, а также памяти к процессору:

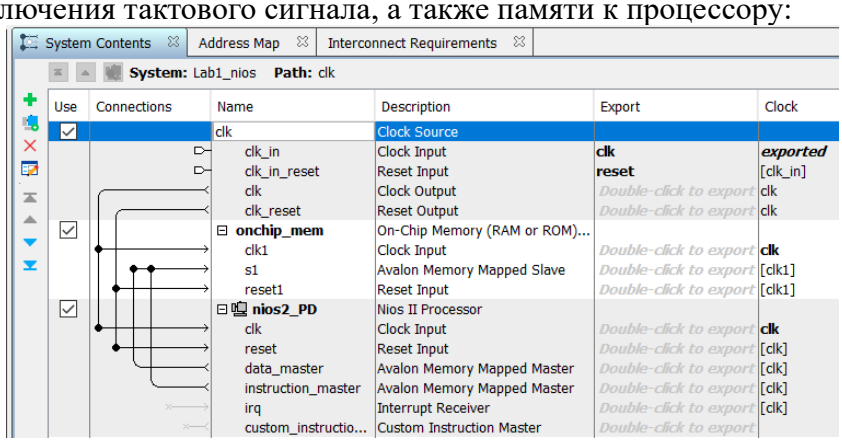


Рис. 3.6. Подключения модулей в проекте.

Конфигурирование процессора NIOS II требует дополнительных настроек. При включении или сбросе устройства происходит событие исключения, сигнализирующее о необходимости начать выполнение процессором с определенного адреса. Этот адрес содержит первоначальную инициализацию процессора. Поскольку сброс может быть выполнен в любой момент, необходимо учесть следующие параметры:

- Местоположение памяти для вектора сброса;
- Местоположение памяти для вектора исключений;

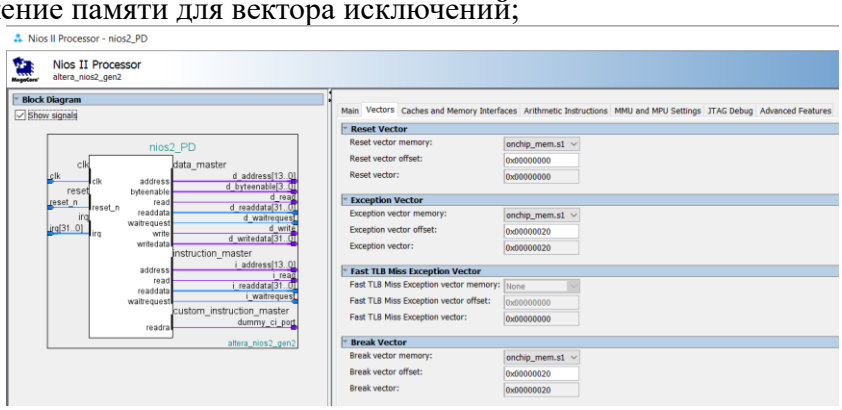


Рис. 3.7. Настройка параметров в модуле nios2_PD.

Теперь добавим модуль для вывода значений на светодиоды:

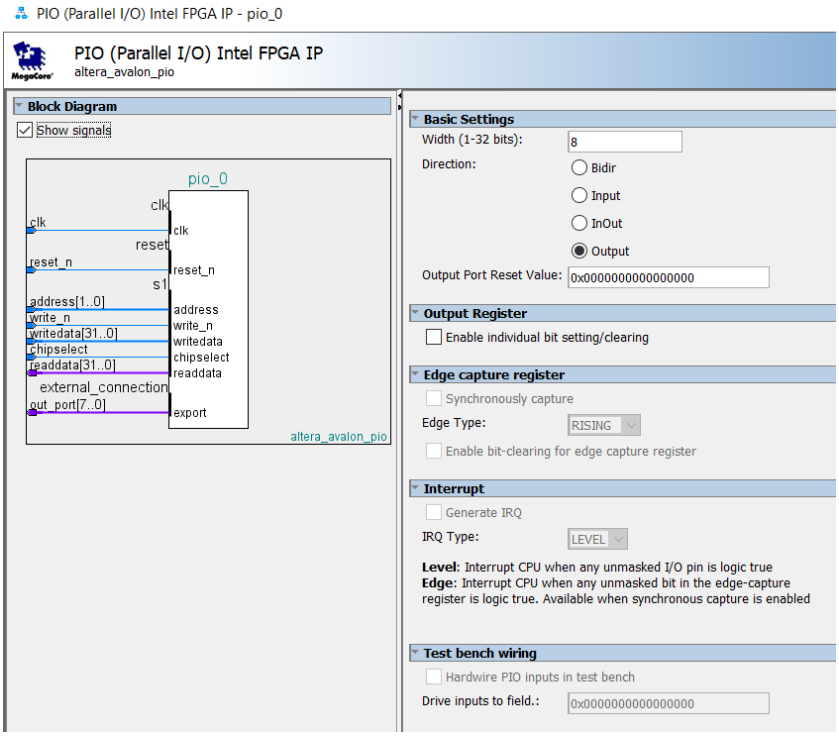


Рис. 3.8. Модуль I/O для светодиодов.

Подключим его к процессору:

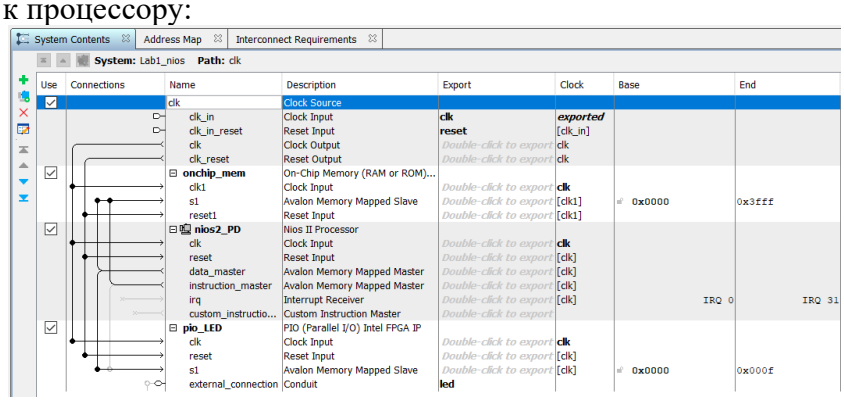


Рис. 3.9. Настройка подключений светодиодов к процессору.

А также таким же образом подключим SW:

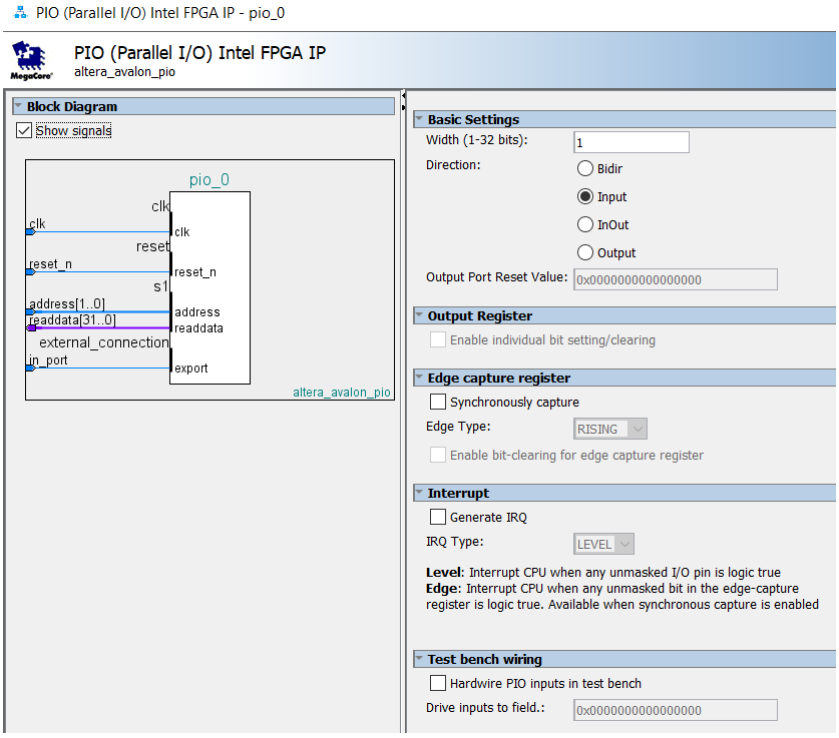


Рис. 3.10. Модуль I/O для переключателей.

И подключим к процессору:

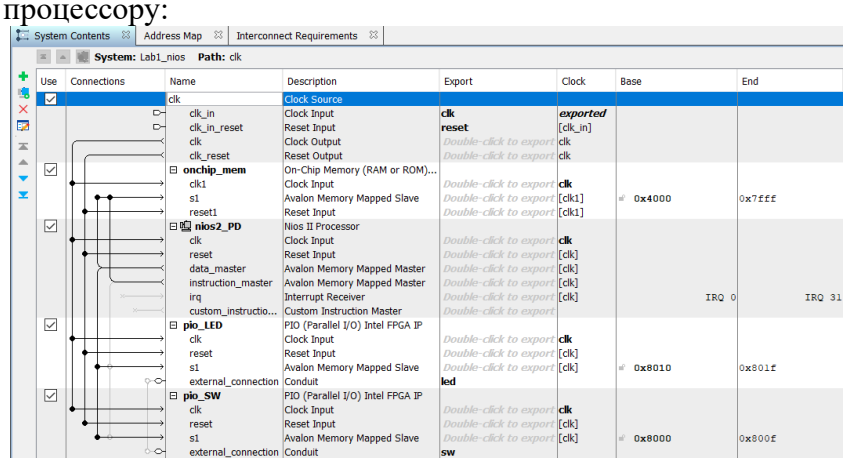


Рис. 3.11. Подключение SW к процессору.

Выполним анализ получившейся системы.

Карта адресов приведена ниже:

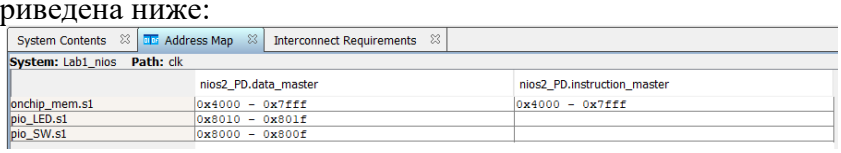


Рис. 3.12. Окно Address Map.

Посмотрим на предустановки системы:

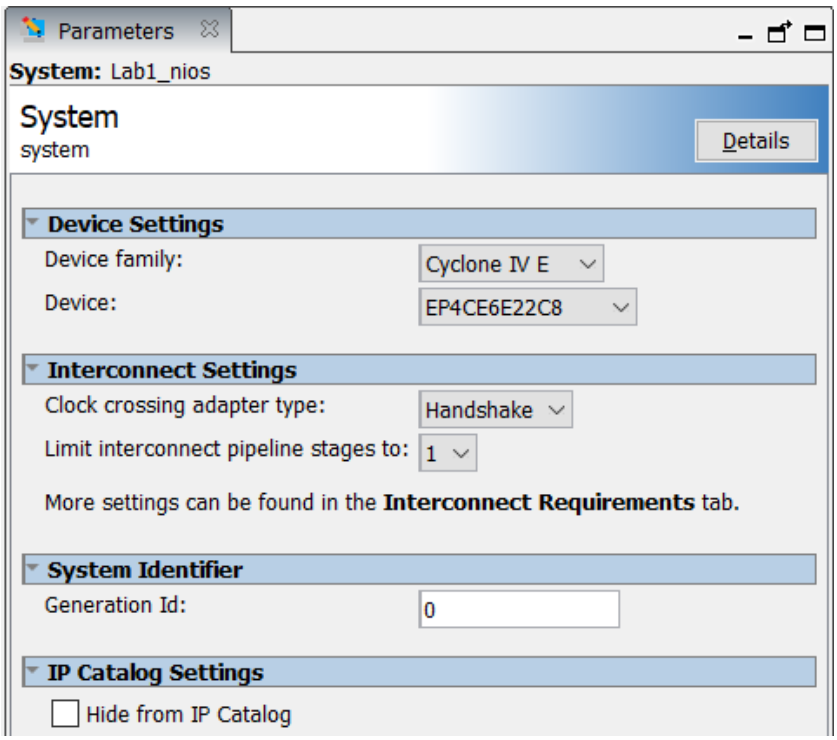


Рис. 3.13. Предустановки системы.

В окне Messages есть только 1 предупреждение, связанное с тем, что не подключён JTAG Debug модуль, однако это было сделано намеренно, поэтому на это предупреждение можем не обращать внимание:

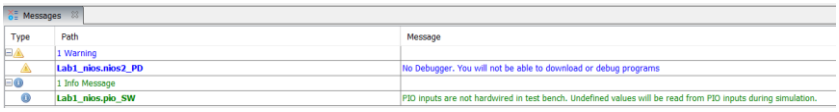


Рис. 3.14. Окно Messages.

Подключим файлы к проекту и создадим модуль верхнего уровня:

```
module Lab1 (
    input bit clk,
    input bit sw,
    input bit pbb,
    output bit [7:0] led
);

    Lab1_nios u0 (
        .clk_clk            (clk),
        .reset_reset_n  (pbb),
        .led_export     (led),
        .sw_export      (sw)
    );

endmodule
```

Выполним компиляцию посмотрим на получившийся результат в RTL Viewer:

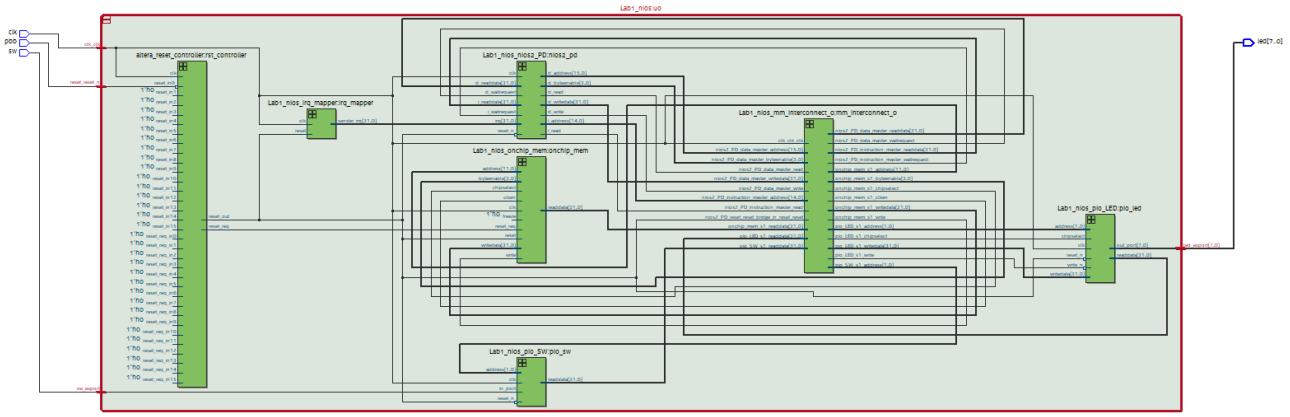


Рис. 3.15. RTL Viewer.

Можем увидеть, что полученная в RTL Viewer схема совпадает с той, что была задана по условию (в зелёном блоке отображается тот фрагмент системы, который был создан средствами PD).

Выполним назначение входов-выходов:

Named: * Edit:

Node Name	Direction	Location	I/O Bank	VREF Group	I/O Standard	Reserved	Current Streng	Slew Rate
clk	Input	PIN_23	1	B1_N0	3.3-V LVTTL		8ma	
led[7]	Output	PIN_65	4	B4_N0	2.5 V		8ma	2 (default)
led[6]	Output	PIN_66	4	B4_N0	2.5 V		8ma	2 (default)
led[5]	Output	PIN_67	4	B4_N0	2.5 V		8ma	2 (default)
led[4]	Output	PIN_68	4	B4_N0	2.5 V		8ma	2 (default)
led[3]	Output	PIN_69	4	B4_N0	2.5 V		8ma	2 (default)
led[2]	Output	PIN_70	4	B4_N0	2.5 V		8ma	2 (default)
led[1]	Output	PIN_71	4	B4_N0	2.5 V		8ma	2 (default)
led[0]	Output	PIN_72	4	B4_N0	2.5 V		8ma	2 (default)
pbb	Input	PIN_58	4	B4_N0	2.5 V		8ma	
sw	Input	PIN_24	2	B2_N0	2.5 V		4ma	
<<new node>>								

Рис. 3.16. Входы-выходы в Pin Planner.

Перейдем к созданию проекта для процессора. Создадим пустой проект в Nios II:

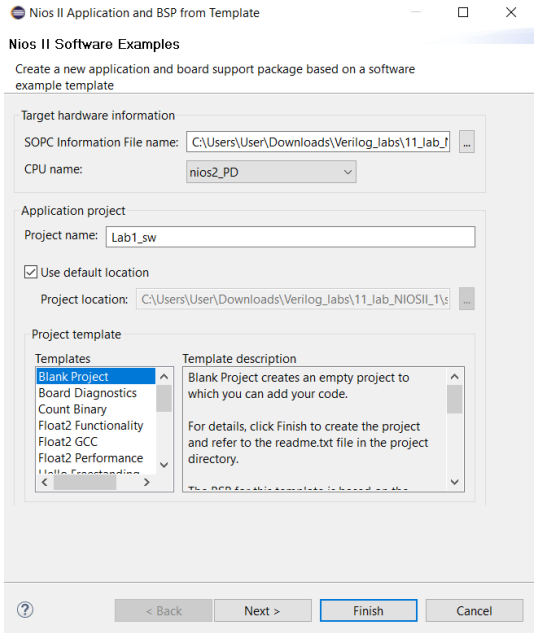


Рис. 3.17. Создание проекта в Nios II.

Далее создадим .c файл с основным кодом проекта:

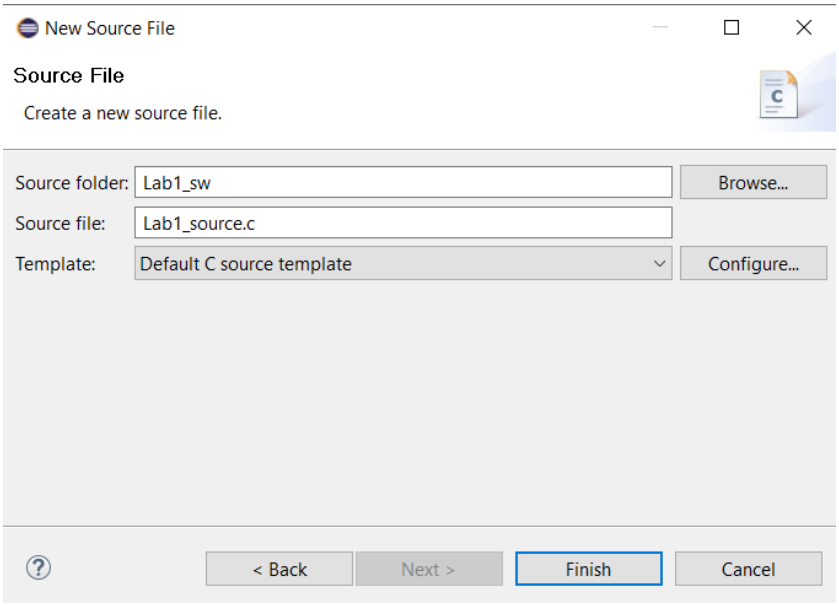


Рис. 3.18. Создание source файла.

Создадим следующий код:

```c
#include "system.h"
#include "altera_avalon_pio_regs.h"
#include <unistd.h>

int main(void)
{
    int sw;
    int count = 255;
    while( 1 )
    {
        usleep (500000);
        sw = IORD_ALTERA_AVALON_PIO_DATA(PIO_SW_BASE);/* read sw[0] value */
        if (sw == 0x1) count++; /* Continue 0-ff counting loop. */
        else count--; /* Continue ff-0 counting loop. */
        IOWR_ALTERA_AVALON_PIO_DATA( PIO_LED_BASE, ~count );
    }
    return 0;
}
```

Каждые пол секунды мы увеличиваем значение счетчика на 1, если sw = 1, иначе уменьшаем значение. Таким образом, используя переключатель мы можем контролировать направление счета.

Выполним сборку проекта:

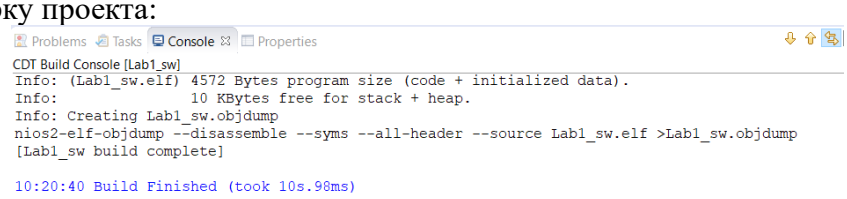


Рис. 3.19. Результат компиляции.

Данный проект занимает 4572 байта, уменьшим объем проекта, выполним следующие настройки:

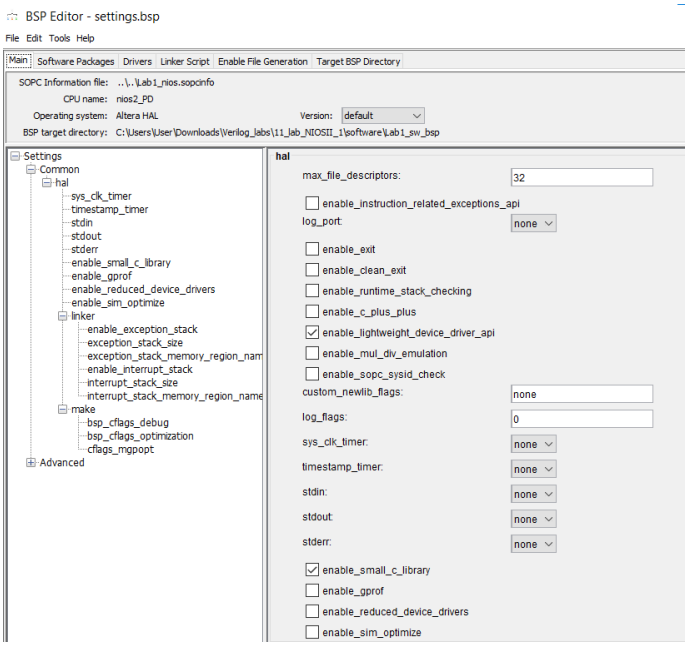


Рис. 3.20. Измеренные настройки генерации.

Выполним повторную компиляцию:

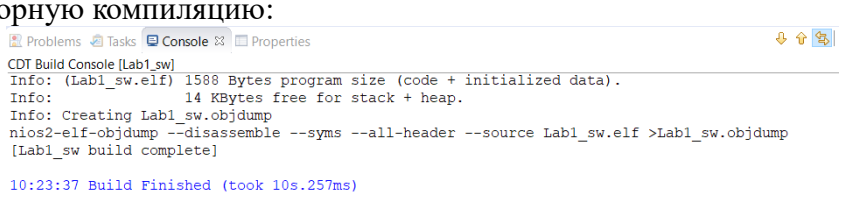


Рис. 3.21. Повторная компиляцию.

Как видим, проект стал занимать много меньше места.

Соберем проект для памяти и добавим в проект.

Проект был загружен на плату, светодиоды LED8-LED1 отображали последовательное увеличение счётчика от 0 до 255 при sw[0] в положении 1, а при переключении sw[0] в положение 0 счёт происходил в обратном порядке с тем же шагом от 255 до 0. Работа на стенде была продемонстрирована преподавателю.

Далее повторим то же самое, установив адреса статически в коде, а также:

```c
#include <unistd.h>

int main(void)
{
    int *psw = (int*) 0x8000;
    int *pled = (int*) 0x8010;
    int count = 64;
    while( 1 )
    {
        usleep (100000);
        if (*psw == 0x1) count++;
        else count--;
        *pled = ~count;
    }
    return 0;
}
```

А также подключим все переключатели, для этого поправим модуль в platform designer:

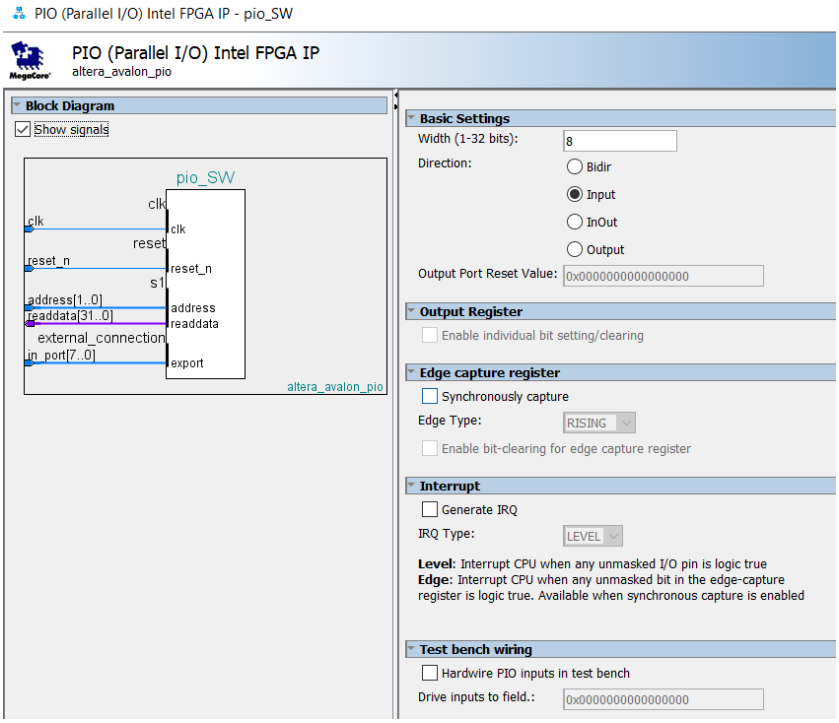
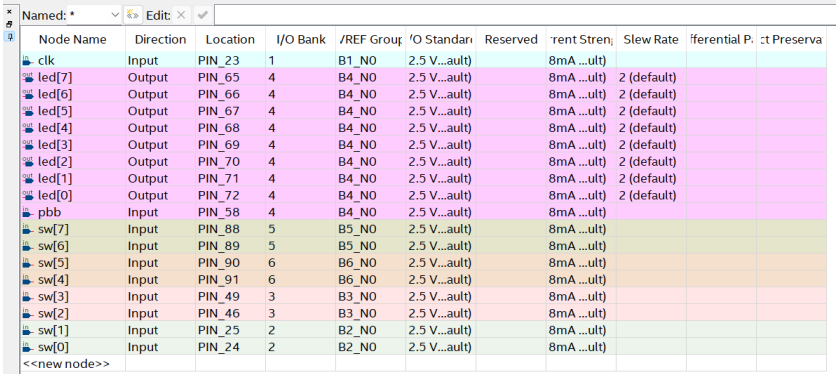


Рис. 3.22. Изменение настроек вводов-выводов.

Также добавим значения в Pin Planner.



Named: * Edit:

Node Name	Direction	Location	I/O Bank	/REF Group	/O Standard	Reserved	rent Stren	Slew Rate	fferential P	t Preserva
clk	Input	PIN_23	1	B1_N0	2.5 V...ault)		8mA ...ult)			
led[7]	Output	PIN_65	4	B4_N0	2.5 V...ault)		8mA ...ult)	2 (default)		
led[6]	Output	PIN_66	4	B4_N0	2.5 V...ault)		8mA ...ult)	2 (default)		
led[5]	Output	PIN_67	4	B4_N0	2.5 V...ault)		8mA ...ult)	2 (default)		
led[4]	Output	PIN_68	4	B4_N0	2.5 V...ault)		8mA ...ult)	2 (default)		
led[3]	Output	PIN_69	4	B4_N0	2.5 V...ault)		8mA ...ult)	2 (default)		
led[2]	Output	PIN_70	4	B4_N0	2.5 V...ault)		8mA ...ult)	2 (default)		
led[1]	Output	PIN_71	4	B4_N0	2.5 V...ault)		8mA ...ult)	2 (default)		
led[0]	Output	PIN_72	4	B4_N0	2.5 V...ault)		8mA ...ult)	2 (default)		
pbb	Input	PIN_58	4	B4_N0	2.5 V...ault)		8mA ...ult)			
sw[7]	Input	PIN_88	5	B5_N0	2.5 V...ault)		8mA ...ult)			
sw[6]	Input	PIN_89	5	B5_N0	2.5 V...ault)		8mA ...ult)			
sw[5]	Input	PIN_90	6	B6_N0	2.5 V...ault)		8mA ...ult)			
sw[4]	Input	PIN_91	6	B6_N0	2.5 V...ault)		8mA ...ult)			
sw[3]	Input	PIN_49	3	B3_N0	2.5 V...ault)		8mA ...ult)			
sw[2]	Input	PIN_46	3	B3_N0	2.5 V...ault)		8mA ...ult)			
sw[1]	Input	PIN_25	2	B2_N0	2.5 V...ault)		8mA ...ult)			
sw[0]	Input	PIN_24	2	B2_N0	2.5 V...ault)		8mA ...ult)			
<<new node>>										

Рис. 3.23. Pin Planner.

Проект был загружен на плату, светодиоды LED8-LED1 отображали последовательное увеличение счётчика от 0 до 255 при sw[0] в положении 1, а при переключении sw[0] в положение 0 счёт происходил в обратном порядке с тем же шагом от 255 до 0. Работа на стенде была продемонстрирована преподавателю.

Далее добавим модуль счета по SW:

```c
#include <unistd.h>

int main(void) {
    char *psw = (char*) 0x8000;
    char *pled = (char*) 0x8010;
    char count = 64;

    while (1) {
        usleep(300000);

        if (((*psw) != 0x00) && (((*psw) - 1) > count))
            count++; /* Continue 0-SW[7:0] counting loop. */
        else
            count = 0; /* start counting loop from 0 */

        *pled = ~count;
    }

    return 0;
}

```

Этот код позволяет выставить по SW верхний предел счета. Устройство было продемонстрировано преподавателю.

4. Вывод:

В процессе выполнения лабораторной работы мы осуществили создание и настройку системы на основе процессора NIOS II с применением программного обеспечения Quartus Prime и интегрированной среды разработки Eclipse. Реализация проекта включала разработку аппаратной составляющей с помощью Platform Designer (PD), написание программного кода в среде разработки Eclipse, где был создан файл для инициализации памяти программ процессора, а также осуществление проекта на плате.

При проверке проекта на плате светодиоды LED8-LED1 отображали пошаговое увеличение счётчика от 0 до 255 при положении sw[0] в единице, а при переключении sw[0] в ноль счёт происходил в обратном порядке с тем же шагом от 255 до 0. Это свидетельствует о правильной работе созданного устройства.

В рамках дополнительных заданий были внедрены различные функциональности, такие как изменение частоты переключения светодиодов, использование указателей для адресации данных и увеличение разрядности ввода-вывода для взаимодействия с переключателями.

Анализ результатов показал успешную работу системы на плате и правильное отображение данных на светодиодах в зависимости от положения переключателей.

Навыки работы с NIOS II, полученные в ходе выполнения этой лабораторной работы, окажутся востребованными при создании проектов в сфере встраиваемых систем, таких как системы управления, обработки сигналов, автоматизации и других.