

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

Отчёт по лабораторной работе № 12

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

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

Руководитель _____ А.П. Антонов
(подпись)

“28” 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: | 9 |

1. Список иллюстраций:

| | |
|---|---|
| Рис. 2.1. Структура проекта. | 3 |
| Рис. 3.1. Создание проекта. | 4 |
| Рис. 3.2. Модуль NIOS II Processor. | 4 |
| Рис. 3.3. Итоговый проект. | 4 |
| Рис. 3.4. Окно Address Map. | 5 |
| Рис. 3.5. Входы-выходы в Pin Planner. | 5 |
| Рис. 3.6. Создание проекта в Nios II. | 6 |
| Рис. 3.7. Создание source файла. | 6 |
| Рис. 3.8. Результат компиляции. | 7 |
| Рис. 3.9. Настройки подключения. | 7 |
| Рис. 3.10. Окно отладки проекта. | 7 |
| Рис. 3.11. Ячейка памяти для светодиодов. | 7 |
| Рис. 3.12. Ячейка памяти для переключателей. | 8 |
| Рис. 3.13. Первая точка остановки. | 8 |
| Рис. 3.14. Память проекта. | 8 |
| Рис. 3.15. Следующая точка остановки. | 8 |
| Рис. 3.16. Память проекта. | 9 |

2. Задание:

2.1. Цель задания:

Расширить знакомство с возможностями по реализации проектов на базе процессора NIOSII:

- 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:

Процессор NIOSII на светодиодах LED8...LED1 отображает двоичные коды чисел от 0 до 255, под управлением данных, получаемых с переключателей SW:

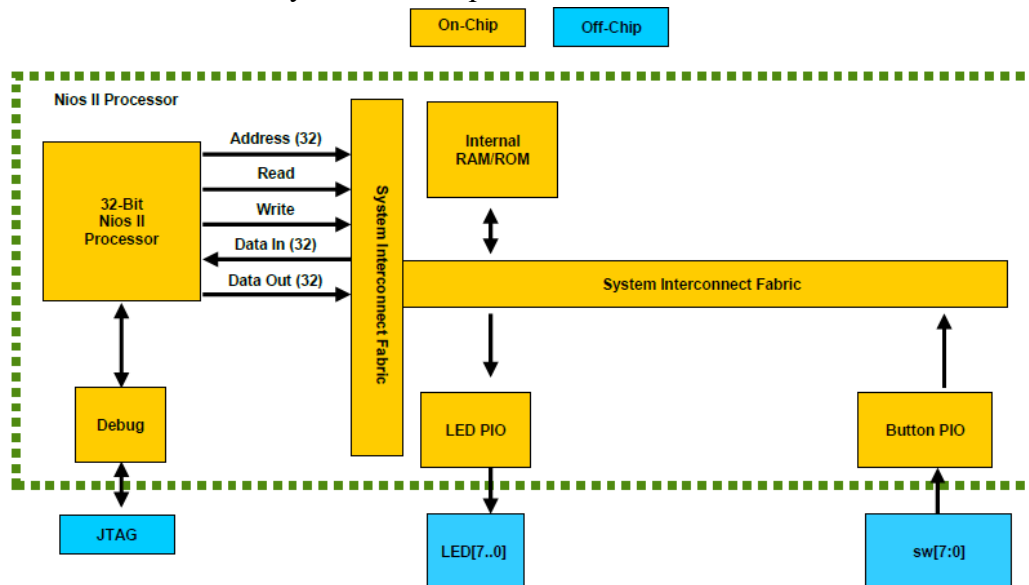


Рис. 2.1. Структура проекта.

Данное устройство содержит 2 модуля I/O, которые используются для обращения к переключателям и светодиодам на плате.

Под управлением процессора NIOSII обеспечивается:

- Опрос состояния переключателя sw[0] (все остальные переключатели в 0).
- Борьба с дребезгом контактов.
- При каждом переключении sw[0] из 1 в 0 - изменение номера включенного светодиода от led1 к led8 на одну позицию (с циклическим переходом от led8 к led1).

3. Ход работы:

Выполним создание проекта со следующими настройками:

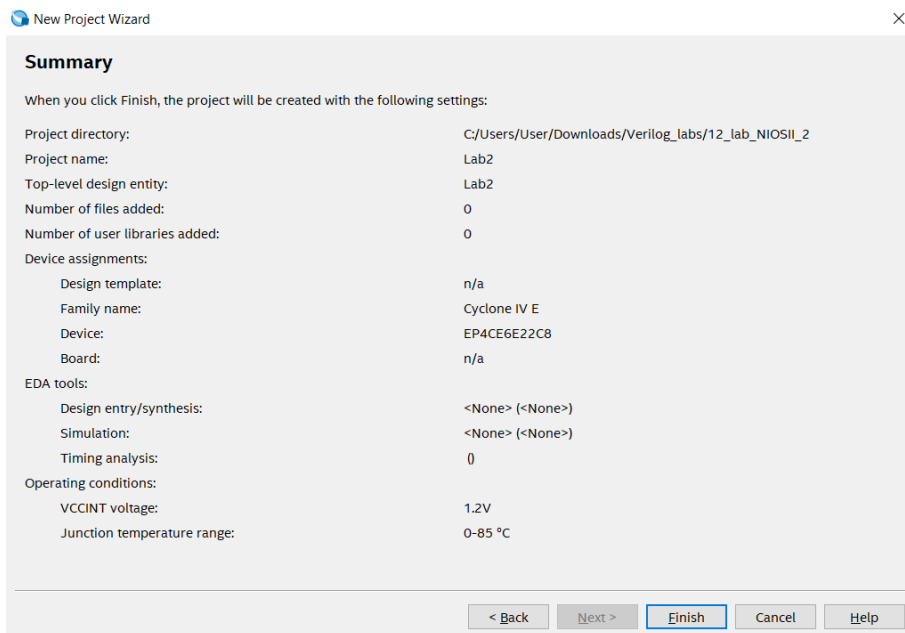


Рис. 3.1. Создание проекта.

Откроем Platform Designer создадим NIOS II Processor модуль и включим для него режим отладки:

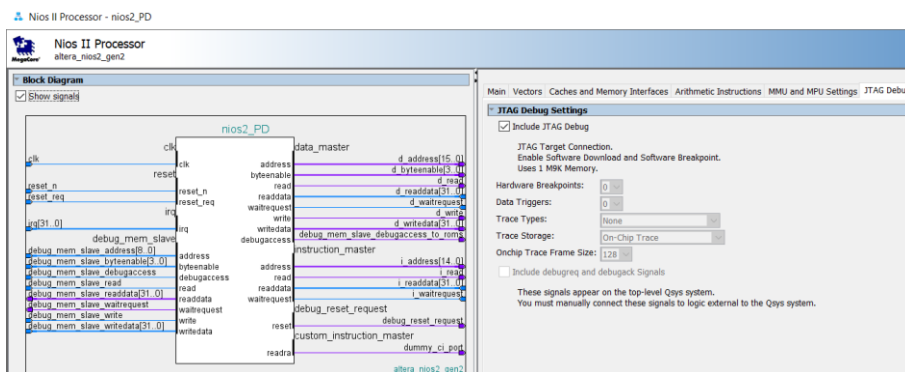


Рис. 3.2. Модуль NIOS II Processor.

Далее выполним стандартную настройку для остальных модулей, в итоге получим следующий проект:

| Use | Connections | Name | Description | Export | Clock | Base | End |
|-------------------------------------|-------------------------------------|----------------------|----------------------------------|------------------------|----------|----------|--------|
| <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | clk | Clock Source | clk | exported | | |
| <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | clk_in | Clock Input | reset | [clk_in] | | |
| <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | clk_in_reset | Reset Input | clk | clk | | |
| <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | clk | Clock Output | Double-click to export | clk | | |
| <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | clk_reset | Reset Output | Double-click to export | clk | | |
| <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | onchip_mem | On-Chip Memory (RAM or ROM)... | Double-click to export | clk | | |
| <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | clk1 | Clock Input | Double-click to export | [clk1] | # 0x4000 | 0x7fff |
| <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | s1 | Avalon Memory Mapped Slave | Double-click to export | [clk1] | | |
| <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | reset1 | Reset Input | Double-click to export | [clk1] | | |
| <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | nios2_PD | Nios II Processor | Double-click to export | clk | | |
| <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | clk | Clock Input | Double-click to export | [clk] | | |
| <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | reset | Reset Input | Double-click to export | [clk] | | |
| <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | data_master | Avalon Memory Mapped Master | Double-click to export | [clk] | | |
| <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | instruction_master | Avalon Memory Mapped Master | Double-click to export | [clk] | | |
| <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | irq | Interrupt Receiver | Double-click to export | [clk] | | IRQ 0 |
| <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | debug_reset_requ... | Reset Output | Double-click to export | [clk] | | |
| <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | debug_mem_slave | Avalon Memory Mapped Slave | Double-click to export | [clk] | # 0x8800 | 0x8fff |
| <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | custom_instructio... | Custom Instruction Master | Double-click to export | [clk] | | |
| <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | pio_LED | PIO (Parallel I/O) Intel FPGA IP | Double-click to export | clk | | |
| <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | clk | Clock Input | Double-click to export | [clk] | | |
| <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | reset | Reset Input | Double-click to export | [clk] | | |
| <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | s1 | Avalon Memory Mapped Slave | Double-click to export | [clk] | # 0x9010 | 0x901f |
| <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | external_connection | Conduit | Double-click to export | led | | |
| <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | pio_SW | PIO (Parallel I/O) Intel FPGA IP | Double-click to export | clk | | |
| <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | clk | Clock Input | Double-click to export | [clk] | | |
| <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | reset | Reset Input | Double-click to export | [clk] | | |
| <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | s1 | Avalon Memory Mapped Slave | Double-click to export | [clk] | # 0x9000 | 0x900f |
| <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | external_connection | Conduit | Double-click to export | sw | | |

Рис. 3.3. Итоговый проект.

Карта адресов приведена ниже:

| System Contents | | | Address Map | | | Interconnect Requirements | | |
|--------------------------|--|--|----------------------|--|--|-----------------------------|--|--|
| System: Lab2_nios | | | Path: clk | | | | | |
| | | | nios2_PD.data_master | | | nios2_PD.instruction_master | | |
| nios2_PD.debug_mem_slave | | | 0x8800 - 0x8fff | | | 0x8800 - 0x8fff | | |
| onchip_mem.s1 | | | 0x4000 - 0x7fff | | | 0x4000 - 0x7fff | | |
| pio_LED.s1 | | | 0x9010 - 0x901f | | | | | |
| pio_SW.s1 | | | 0x9000 - 0x900f | | | | | |

Рис. 3.4. Окно Address Map.

Выполним build разработанного проекта.

Подключим файлы к проекту и создадим модуль верхнего уровня:

```

Verilog_labs - Lab2.v

1  module Lab2 (
2      input bit clk,
3      input bit [7:0] sw,
4      input bit pbb,
5      output bit [7:0] led
6  );
7
8  Lab2_nios u0 (
9      .clk_clk(clk),
10     .reset_reset_n(pbb),
11     .led_export(led),
12     .sw_export(sw)
13 );
14
15 endmodule

```

Выполним назначение входов-выходов:

| Node Name | Direction | Location | I/O Bank | VREF Group | I/O Standard | Current Strength | Slew Rate | Reserved | Differential Pair | Strict Preservation |
|-----------|-----------|----------|----------|------------|--------------|------------------|-------------|----------|-------------------|---------------------|
| clk | Input | PIN_23 | 1 | B1_NO | 3.3-V LVTTTL | 8mA (default) | | | | |
| led[7] | Output | PIN_65 | 4 | B4_NO | 2.5 V | 8mA | 2 (default) | | | |
| led[6] | Output | PIN_66 | 4 | B4_NO | 2.5 V | 8mA | 2 (default) | | | |
| led[5] | Output | PIN_67 | 4 | B4_NO | 2.5 V | 8mA | 2 (default) | | | |
| led[4] | Output | PIN_68 | 4 | B4_NO | 2.5 V | 8mA | 2 (default) | | | |
| led[3] | Output | PIN_69 | 4 | B4_NO | 2.5 V | 8mA | 2 (default) | | | |
| led[2] | Output | PIN_70 | 4 | B4_NO | 2.5 V | 8mA | 2 (default) | | | |
| led[1] | Output | PIN_71 | 4 | B4_NO | 2.5 V | 8mA | 2 (default) | | | |
| led[0] | Output | PIN_72 | 4 | B4_NO | 2.5 V | 8mA | 2 (default) | | | |
| pbb | Input | PIN_58 | 4 | B4_NO | 2.5 V | 8mA (default) | | | | |
| sw[7] | Input | PIN_88 | 5 | B5_NO | 3.3-V LVTTTL | 8mA (default) | | | | |
| sw[6] | Input | PIN_89 | 5 | B5_NO | 3.3-V LVTTTL | 8mA (default) | | | | |
| sw[5] | Input | PIN_90 | 6 | B6_NO | 3.3-V LVTTTL | 8mA (default) | | | | |
| sw[4] | Input | PIN_91 | 6 | B6_NO | 3.0-V LVTTTL | 8mA (default) | | | | |
| sw[3] | Input | PIN_49 | 3 | B3_NO | 3.0-V LVTTTL | 8mA (default) | | | | |
| sw[2] | Input | PIN_46 | 3 | B3_NO | 3.3-V LVTTTL | 8mA (default) | | | | |
| sw[1] | Input | PIN_25 | 2 | B2_NO | 3.3-V LVTTTL | 8mA (default) | | | | |
| sw[0] | Input | PIN_24 | 2 | B2_NO | 3.3-V LVTTTL | 8mA (default) | | | | |

Рис. 3.5. Входы-выходы в Pin Planner.

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

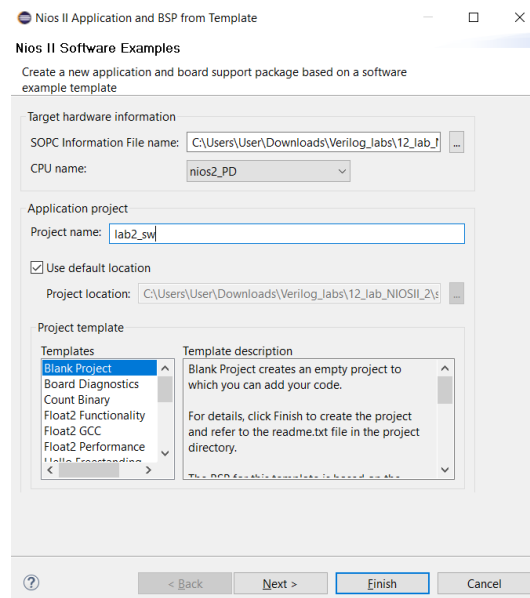


Рис. 3.6. Создание проекта в Nios II.

Далее создадим .c файл с основным кодом проекта:

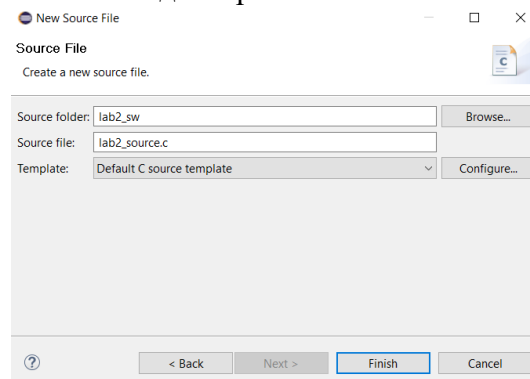


Рис. 3.7. Создание source файла.

Создадим следующий код:

```
Verilog_labs - lab2_source.c
1  #include "system.h"
2  #include "altera_avalon_pio_regs.h"
3  #include <unistd.h>
4
5  #define EQ_ONE 0x01
6  #define DEBOUNCE 30000
7
8  int main(void) {
9      int sw;
10     int led = 0x00;
11
12     while (1) {
13         sw = IORD_ALTERA_AVALON_PIO_DATA(PIO_SW_BASE);
14
15         if (sw != EQ_ONE) {
16             if (led >= 0x80 || led == 0x00)
17                 led = 0x01;
18             else
19                 led = led << 1;
20             IOWR_ALTERA_AVALON_PIO_DATA(PIO_LED_BASE, ~led);
21
22             usleep(DEBOUNCE);
23             while (sw != EQ_ONE)
24                 sw = IORD_ALTERA_AVALON_PIO_DATA(PIO_SW_BASE);
25             usleep(DEBOUNCE);
26         }
27     }
28 }
```

Он считывает значение с счетчика и если значение равно 0x01 то, выполняет сдвиг горящего светодиода влево, выводит это значение на непосредственно I/O, после чего засыпает на 30000 ms, что позволяет пропустить основной дребезг контакта, после чего дожидается, когда переключатель вернется в 0 и опять засыпает на тоже самое время.

Выполним сборку проекта:

```

CDT Build Console [lab2_sw]
Info: (lab2_sw.elf) 4616 Bytes program size (code + initialized data).
Info: 10 KBytes free for stack + heap.
Info: Creating lab2_sw.objdump
nios2-elf-objdump --disassemble --syms --all-header --source lab2_sw.elf
[lab2_sw build complete]

13:32:55 Build Finished (took 13s.291ms)

```

Рис. 3.8. Результат компиляции.

В отличие от предыдущей лабораторной, выполним отладку разработанной программы на плате, для этого заливаем разработанную программу в Quartus на плату, а в Эклипсе настраиваем подключение к процессору:

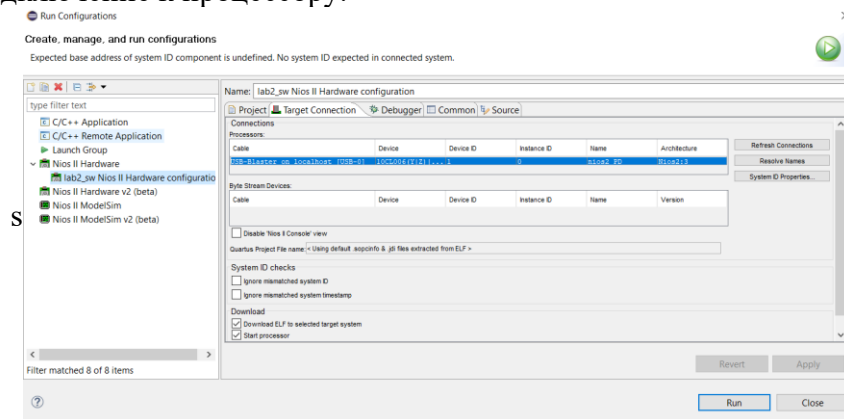


Рис. 3.9. Настройки подключения.

После чего переходим в режим отладки, он будет выглядеть следующим образом:

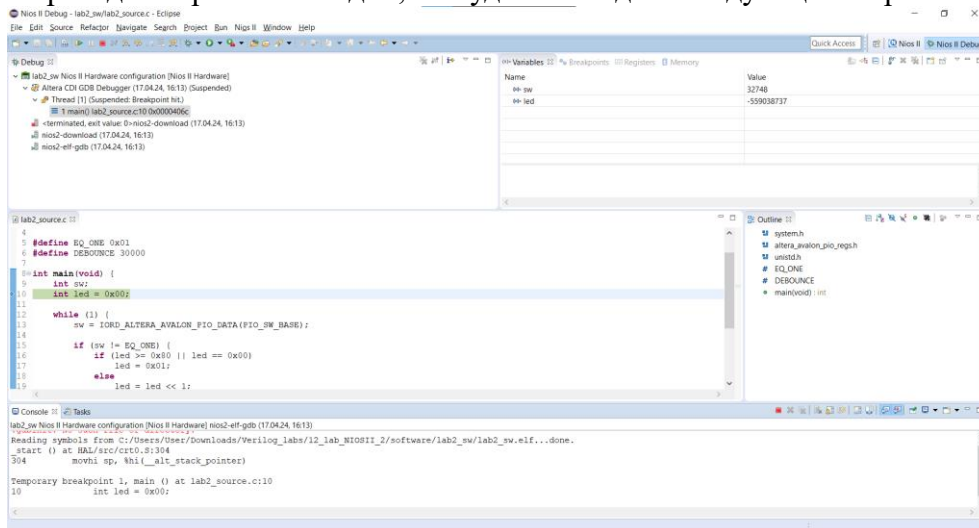


Рис. 3.10. Окно отладки проекта.

В окне memory мы можем видеть используемые ячейки памяти проекта:

| Address | 0 - 3 | 4 - 7 | 8 - B | C - F |
|----------|----------|----------|----------|----------|
| 00007FE0 | F0400000 | EC7F0000 | EFBEADDE | F87F0000 |
| 00007FF0 | 2C420000 | F87F0000 | EFBEADDE | 58400000 |
| 00008000 | 34004000 | 14085008 | 3A680008 | 00000000 |
| 00008010 | 00000000 | 00000000 | 00000000 | 00000000 |
| 00008020 | 3400C006 | 1400E0DE | 34008006 | 1476B4D6 |
| 00008030 | 34008000 | 14F99510 | 3400C000 | 14FDD518 |
| 00008040 | 2603C010 | 15000010 | 04018010 | 36FDF110 |
| 00008050 | C0150400 | 001E0400 | 06FF3F00 | 04FCFFDE |
| 00008060 | 1503C0DF | 150200DF | 040200DF | 3AA03D00 |

Рис. 3.11. Ячейка памяти для светодиодов.

| Address | 0 - 3 | 4 - 7 | 8 - B | C - F |
|----------|----------|----------|----------|----------|
| 00007FE0 | F0400000 | EC7F0000 | EFBEADDE | F87F0000 |
| 00007FF0 | 2C420000 | F87F0000 | EFBEADDE | 58400000 |
| 00008000 | 34004000 | 14085008 | 3A680008 | 00000000 |
| 00008010 | 00000000 | 00000000 | 00000000 | 00000000 |
| 00008020 | 3400C006 | 1400E0DE | 34008006 | 1476B4D6 |
| 00008030 | 34008000 | 14F99510 | 3400C000 | 14FDD518 |
| 00008040 | 2603C010 | 15000010 | 04018010 | 36FDFF10 |
| 00008050 | C0150400 | 001E0400 | 06FF3F00 | 04FCFFDE |
| 00008060 | 1503C0DF | 150200DF | 040200DF | 3AA03D00 |

Рис. 3.12. Ячейка памяти для переключателей.

Добавим точки останова в строках 22 и 25, это позволит посмотреть на значения переменных прямо во время попадания проекта в эти места.

Переключаем sw из 1 в 0 и получаем следующий результат:

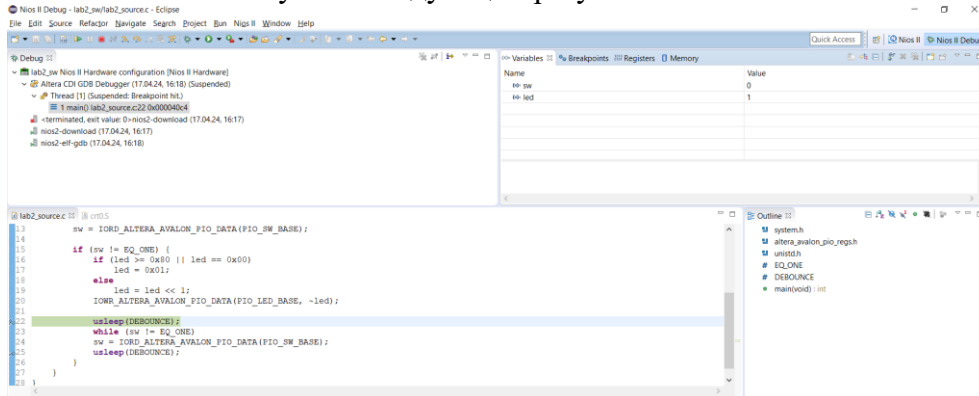


Рис. 3.13. Первая точка остановки.

Значения в адресах выглядят следующим образом:

| Address | 0 - 3 | 4 - 7 | 8 - B | C - F |
|----------|----------|----------|----------|----------|
| 00007FE0 | F0400000 | 00000000 | 01000000 | F87F0000 |
| 00007FF0 | 2C420000 | F87F0000 | EFBEADDE | 58400000 |
| 00008000 | 34004000 | 14085008 | 3A680008 | 00000000 |
| 00008010 | 00000000 | 00000000 | 00000000 | 00000000 |
| 00008020 | 3400C006 | 1400E0DE | 34008006 | 1476B4D6 |
| 00008030 | 34008000 | 14F99510 | 3400C000 | 14FDD518 |
| 00008040 | 2603C010 | 15000010 | 04018010 | 36FDFF10 |
| 00008050 | C0150400 | 001E0400 | 06FF3F00 | 04FCFFDE |
| 00008060 | 1503C0DF | 150200DF | 040200DF | 15FF3FE0 |

Рис. 3.14. Память проекта.

Для попадания в следующую точку остановки выполним переключение из sw обратно из 0 в 1:

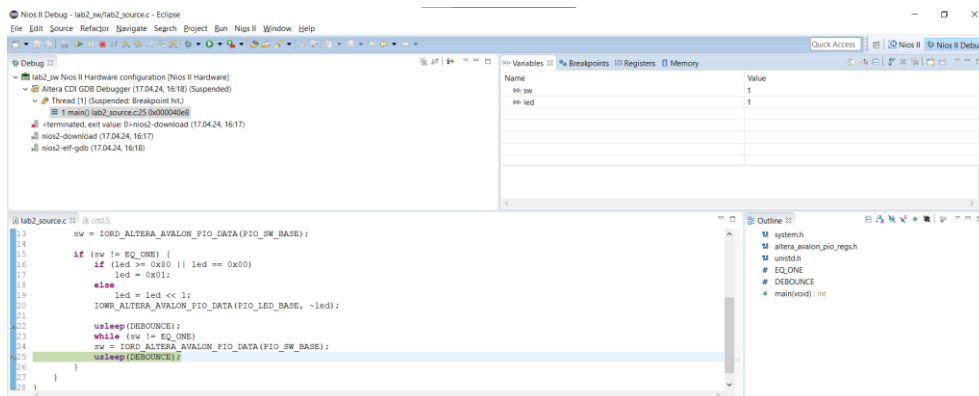


Рис. 3.15. Следующая точка остановки.

Память изменится следующим образом:

| Address | 0 - 3 | 4 - 7 | 8 - B | C - F |
|----------|----------|----------|----------|-----------|
| 00007FE0 | CC400000 | 01000000 | 01000000 | F87F0000 |
| 00007FF0 | 2C420000 | F87F0000 | EFBEADDE | 58400000 |
| 00008000 | 34004000 | 14085008 | 3A680008 | 00000000 |
| 00008010 | 00000000 | 00000000 | 00000000 | 00000000 |
| 00008020 | 3400C006 | 1400E0DE | 34008006 | 1476B4D6 |
| 00008030 | 34008000 | 14F99510 | 3400C000 | 14FDD518 |
| 00008040 | 2603C010 | 15000010 | 04018010 | 36FDDFF10 |
| 00008050 | C0150400 | 001E0400 | 06FF3F00 | 04FCFFDE |
| 00008060 | 1503C0DF | 150200DF | 040200DF | 15FF3FE0 |

Рис. 3.16. Память проекта.

Работа на стенде была продемонстрирована преподавателю.

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

```

Verilog_labs - lab2_source.c
1  #include "system.h"
2  #include "altera_avalon_pio_regs.h"
3  #include <unistd.h>
4
5  #define EQ_ONE 0x01
6  #define DEBOUNCE 30000
7
8  int main(void) {
9      int* psw = (int*) 0x9000;
10     int* led_out = (int*) 0x9010;
11     int led = 0x80;
12     *led_out = 0xF0;
13     while (1) {
14
15         if (*psw != EQ_ONE) {
16             if (led > 0x80 || led == 0x01)
17                 led = 0x80;
18             else
19                 led = led >> 1;
20             *led_out = led;
21
22             usleep(DEBOUNCE);
23             while (*(psw) != EQ_ONE);
24             usleep(DEBOUNCE);
25         }
26     }
27 }

```

Теперь переход осуществляется по абсолютным адресам. Также, была изменена та часть кода, которая отвечала за сдвиг светодиода. Теперь сдвиг происходит не влево, а вправо. Работа на стенде была продемонстрирована преподавателю.

4. Вывод:

При проведении лабораторной работы была осуществлена настройка системы на основе процессора NIOS II с использованием пакета Quartus Prime и Eclipse IDE. Для этого была использована система, разработанная на предыдущей лабораторной работе. Реализация включала создание аппаратной части с помощью Platform Designer, программной части с использованием среды разработки Eclipse, создание файла инициализации для модуля памяти программ процессора и обработку проекта на плате с помощью Eclipse.

При тестировании проекта на плате светодиоды отображали последовательный сдвиг влево, который происходил при переключении SW[0] из 1 в 0. Дополнительно были реализованы функциональности изменения направления сдвига и использование указателей для адресации данных. Проверка работы дополнительного задания была успешно выполнена преподавателем. Анализ результатов показал успешную работу системы на плате и корректное отображение данных на светодиодах в зависимости от положения переключателя SW[0].

Опыт работы с NIOS II может быть полезен при создании проектов в области встраиваемых систем, таких как системы управления, обработки сигналов и автоматизации. Среда Eclipse облегчает процесс отладки и отслеживание адресов в реальном времени.