САНКТ-ПЕТЕРБУРГСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

ПЕТРА ВЕЛИКОГО

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Институт компьютерных наук и технологий

Высшая школа интеллектуальных систем и суперкомпьютерных технологий

Отчет

по заданию lab 15

Дисциплина

«Технологии проектирования аппаратных средств компьютерных систем»

выполнил:

Курякин Д. А

группа:

преподаватель:

Антонов А. П.

Санкт-Петербург

2022

Оглавление

[1 Задание labn\_2 3](#_Toc105855331)

[1.1 Задание 3](#_Toc105855332)

[1.2 Создание макета в Platform Designer 3](#_Toc105855333)

[1.3 Анализ системы 4](#_Toc105855334)

[1.4 Программирование Nios 2 7](#_Toc105855335)

[1.5 Выводы 8](#_Toc105855336)

[2 Задание labn\_2s 8](#_Toc105855337)

[2.1 Задание 8](#_Toc105855338)

[2.2 Создание макета в Platform Designer 9](#_Toc105855339)

[2.3 Анализ системы 10](#_Toc105855340)

[2.4 Программирование Nios 2 13](#_Toc105855341)

[2.5 Выводы 14](#_Toc105855342)

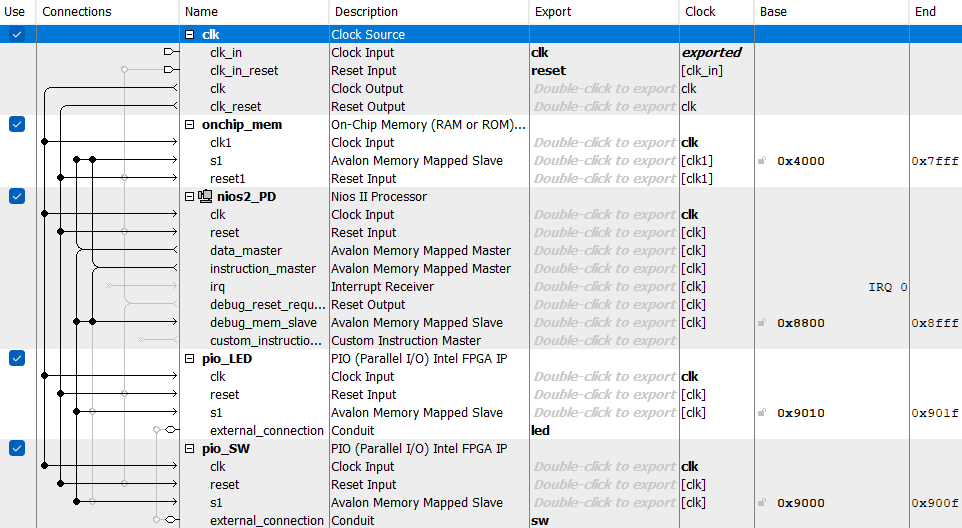
# Задание labn\_2

## Задание

Выполнить шаги из примера в методических указаниях.

## Создание макета в Platform Designer

Созданный макет приведен на рис. 1-1.



*Рис. 1-1. Макет в Platform Designer*

Файл верхнего уровня приведен в листинге 1-1.

Листинг 1-1. Lab2.sv

|  |
| --- |
| module Lab2(  input bit clk,  input bit sw,  input bit pbb,  output bit [7:0] led  );  Lab2\_nios u0(  .clk\_clk (clk), // clk.clk  .reset\_reset\_n (pbb), // reset.reset\_n  .led\_export (led), // led.export  .sw\_export (sw) // sw.export  );  endmodule |

Файл описания временных параметров приведен в листинге 1-2.

Листинг 1-2.

|  |
| --- |
| set\_time\_format -unit ns -decimal\_places 3  create\_clock -name {clock} -period 40.000 -waveform { 0.000 20.000 } [get\_ports {clk}]  derive\_clock\_uncertainty  set\_input\_delay -add\_delay -clock [get\_clocks {clock}] 10.000 [get\_ports {pbb}]  set\_input\_delay -add\_delay -clock [get\_clocks {clock}] 10.000 [get\_ports {sw}]  set\_input\_delay -add\_delay -clock [get\_clocks {clock}] 10.000 [get\_ports {altera\_reserved\_tdi}]  set\_input\_delay -add\_delay -clock [get\_clocks {clock}] 10.000 [get\_ports {altera\_reserved\_tms}]  set\_output\_delay -add\_delay -clock [get\_clocks {clock}] 10.000 [get\_ports {led[0]}]  set\_output\_delay -add\_delay -clock [get\_clocks {clock}] 10.000 [get\_ports {led[1]}]  set\_output\_delay -add\_delay -clock [get\_clocks {clock}] 10.000 [get\_ports {led[2]}]  set\_output\_delay -add\_delay -clock [get\_clocks {clock}] 10.000 [get\_ports {led[3]}]  set\_output\_delay -add\_delay -clock [get\_clocks {clock}] 10.000 [get\_ports {led[4]}]  set\_output\_delay -add\_delay -clock [get\_clocks {clock}] 10.000 [get\_ports {led[5]}]  set\_output\_delay -add\_delay -clock [get\_clocks {clock}] 10.000 [get\_ports {led[6]}]  set\_output\_delay -add\_delay -clock [get\_clocks {clock}] 10.000 [get\_ports {led[7]}]  set\_input\_delay -add\_delay -clock [get\_clocks {clock}] 10.000 [get\_ports {altera\_reserved\_tdo}] |

## Анализ системы

На рис. 1-3 – 1-6 приведен анализ системы созданной в Platform Designer.

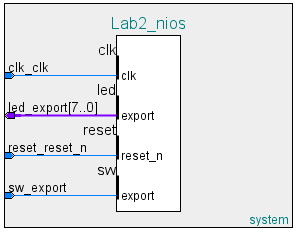


Рис.1-3. Block Symbol

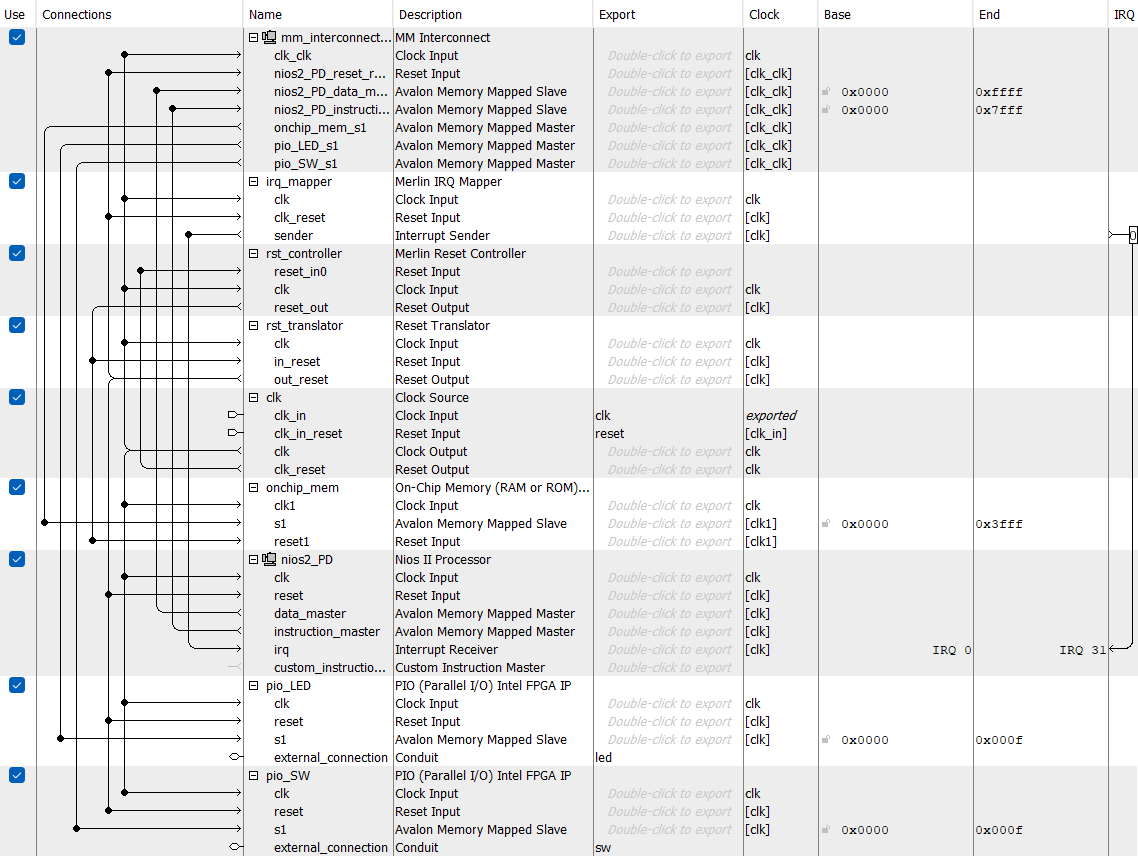


Рис.1-4. Show System with Platform Designer Interconnect

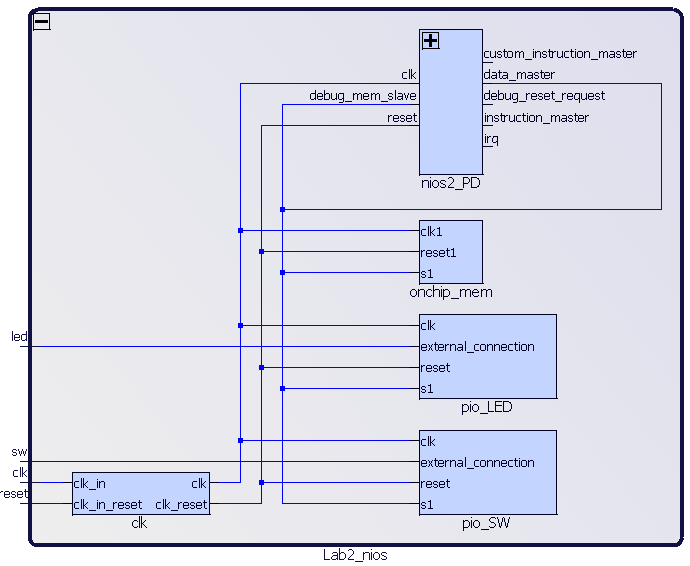


Рис.1-5. Schematic. In

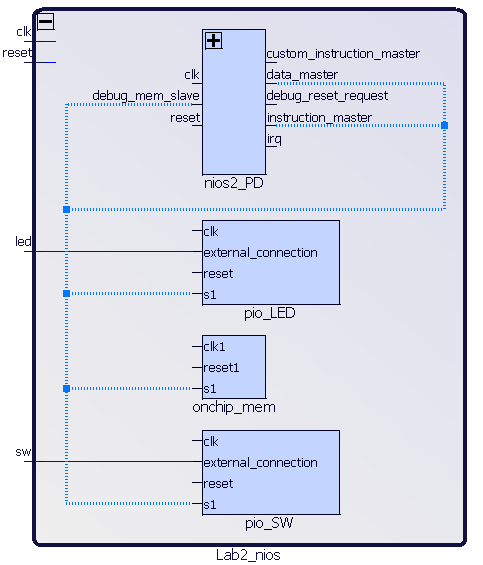


Рис.1-6. Schematic. clk

## Программирование Nios 2

Программы, написанные для soft-процессора Nios 2 представлены в листингах 1-3 – 1-4.

Листинг 1-3. Программа для Nios 2 с использованием переменных.

|  |
| --- |
| **#include** "system.h"  **#include** "altera\_avalon\_pio\_regs.h"  **#include** <unistd.h>  **#define** EQ\_ONE 0x1  **#define** DEBOUNCE 30000 // Time in microseconds to wait for switch debounce  **int** **main**(**void**) {  **int** sw ; // Use to hold sw value  **int** led = 0x00; // Use to write to led  **while** (1)  { // Read buttons via pio  sw = IORD\_ALTERA\_AVALON\_PIO\_DATA(PIO\_SW\_BASE);  **if** (sw != EQ\_ONE) // if value of sw is 0  {  **if** (led >= 0x80 || led==0x00)  led = 0x01; // loop  **else**  led = led << 1; // shift left on board (led0 is far right)  IOWR\_ALTERA\_AVALON\_PIO\_DATA(PIO\_LED\_BASE,~led); // Write new value to pio  // Switch debounce routine  **usleep** (DEBOUNCE);  **while** (sw != EQ\_ONE) // wait for sw =1  sw = IORD\_ALTERA\_AVALON\_PIO\_DATA(PIO\_SW\_BASE);  **usleep** (DEBOUNCE);  }  }  } |

Листинг 1-4. Программа для Nios 2 с использованием адресов.

|  |
| --- |
| **#include** "system.h"  **#include** "altera\_avalon\_pio\_regs.h"  **#include** <unistd.h>  **#define** EQ\_ONE 0x1  **#define** DEBOUNCE 30000 // Time in microseconds to wait for switch debounce  **int** **main**(**void**) {  **int** \*psw = (**int**\*) 0x9000;; // Use to hold sw value  **int** \*pled = (**int**\*) 0x9010; // Use to write to led  **while** (1)  { // Read buttons via pio  **if** (\*psw != EQ\_ONE) // if value of sw is 0  {  **if** (\*pled >= 0x80 || \*pled==0x00)  \*pled = 0x01; // loop  **else**  \*pled = \*pled << 1; // shift left on board (led0 is far right)  IOWR\_ALTERA\_AVALON\_PIO\_DATA(PIO\_LED\_BASE,~led); // Write new value to pio  // Switch debounce routine  **usleep** (DEBOUNCE);  **while** (\*psw != EQ\_ONE) // wait for sw =1  \*psw = 0x9000;  **usleep** (DEBOUNCE);  }  }  } |

## Выводы

В результате были пройдены шаги из презентации для создания макета в Platform Designer и написания программы для soft-процессора Nios 2.

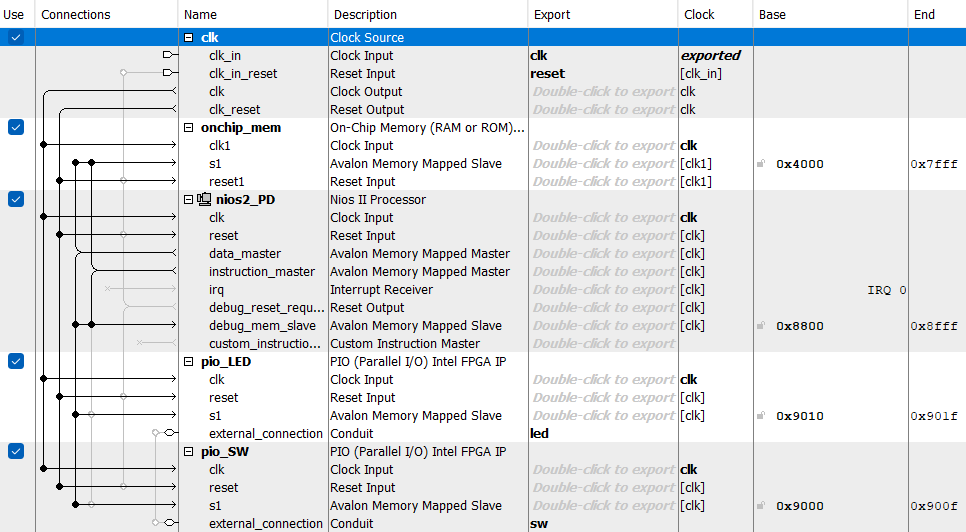
# Задание labn\_2s

## Задание

Выполнить шаги из примера в методических указаниях.

## Создание макета в Platform Designer

Созданный макет приведен на рис. 2-1.



*Рис. 2-1. Макет в Platform Designer*

Файл верхнего уровня приведен в листинге 2-1.

Листинг 2-1. Lab1.sv

|  |
| --- |
| module Lab2s\_nios\_inst(  input bit clk,  output bit [7:0] led,  input bit reset,  input bit [7:0] sw  );  Lab2\_nios inst(  .clk\_clk(clk), // clk.clk  .led\_export(led), // led.export  .reset\_reset\_n(reset), // reset.reset\_n  .sw\_export(sw) // sw.export  );  endmodule |

Файл описания временных параметров приведен в листинге 1-2.

Листинг 2-2.

|  |
| --- |
| set\_time\_format -unit ns -decimal\_places 3  create\_clock -name {clock} -period 40.000 -waveform { 0.000 20.000 } [get\_ports {clk}]  set\_clock\_uncertainty -rise\_from [get\_clocks {clock}] -rise\_to [get\_clocks {clock}] 0.020  set\_clock\_uncertainty -rise\_from [get\_clocks {clock}] -fall\_to [get\_clocks {clock}] 0.020  set\_clock\_uncertainty -fall\_from [get\_clocks {clock}] -rise\_to [get\_clocks {clock}] 0.020  set\_clock\_uncertainty -fall\_from [get\_clocks {clock}] -fall\_to [get\_clocks {clock}] 0.020  set\_input\_delay -add\_delay -clock [get\_clocks {clock}] 10.000 [get\_ports {reset}]  set\_input\_delay -add\_delay -clock [get\_clocks {clock}] 10.000 [get\_ports {sw[0]}]  set\_input\_delay -add\_delay -clock [get\_clocks {clock}] 10.000 [get\_ports {sw[1]}]  set\_input\_delay -add\_delay -clock [get\_clocks {clock}] 10.000 [get\_ports {sw[2]}]  set\_input\_delay -add\_delay -clock [get\_clocks {clock}] 10.000 [get\_ports {sw[3]}]  set\_input\_delay -add\_delay -clock [get\_clocks {clock}] 10.000 [get\_ports {sw[4]}]  set\_input\_delay -add\_delay -clock [get\_clocks {clock}] 10.000 [get\_ports {sw[5]}]  set\_input\_delay -add\_delay -clock [get\_clocks {clock}] 10.000 [get\_ports {sw[6]}]  set\_input\_delay -add\_delay -clock [get\_clocks {clock}] 10.000 [get\_ports {sw[7]}]  set\_output\_delay -add\_delay -clock [get\_clocks {clock}] 10.000 [get\_ports {led[0]}]  set\_output\_delay -add\_delay -clock [get\_clocks {clock}] 10.000 [get\_ports {led[1]}]  set\_output\_delay -add\_delay -clock [get\_clocks {clock}] 10.000 [get\_ports {led[2]}]  set\_output\_delay -add\_delay -clock [get\_clocks {clock}] 10.000 [get\_ports {led[3]}]  set\_output\_delay -add\_delay -clock [get\_clocks {clock}] 10.000 [get\_ports {led[4]}]  set\_output\_delay -add\_delay -clock [get\_clocks {clock}] 10.000 [get\_ports {led[5]}]  set\_output\_delay -add\_delay -clock [get\_clocks {clock}] 10.000 [get\_ports {led[6]}]  set\_output\_delay -add\_delay -clock [get\_clocks {clock}] 10.000 [get\_ports {led[7]}] |

## Анализ системы

На рис. 1-3 – 1-6 приведен анализ системы созданной в Platform Designer.

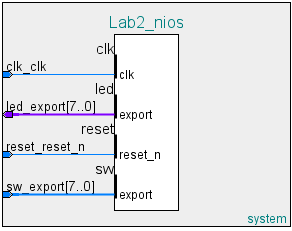


Рис.1-3. Block Symbol

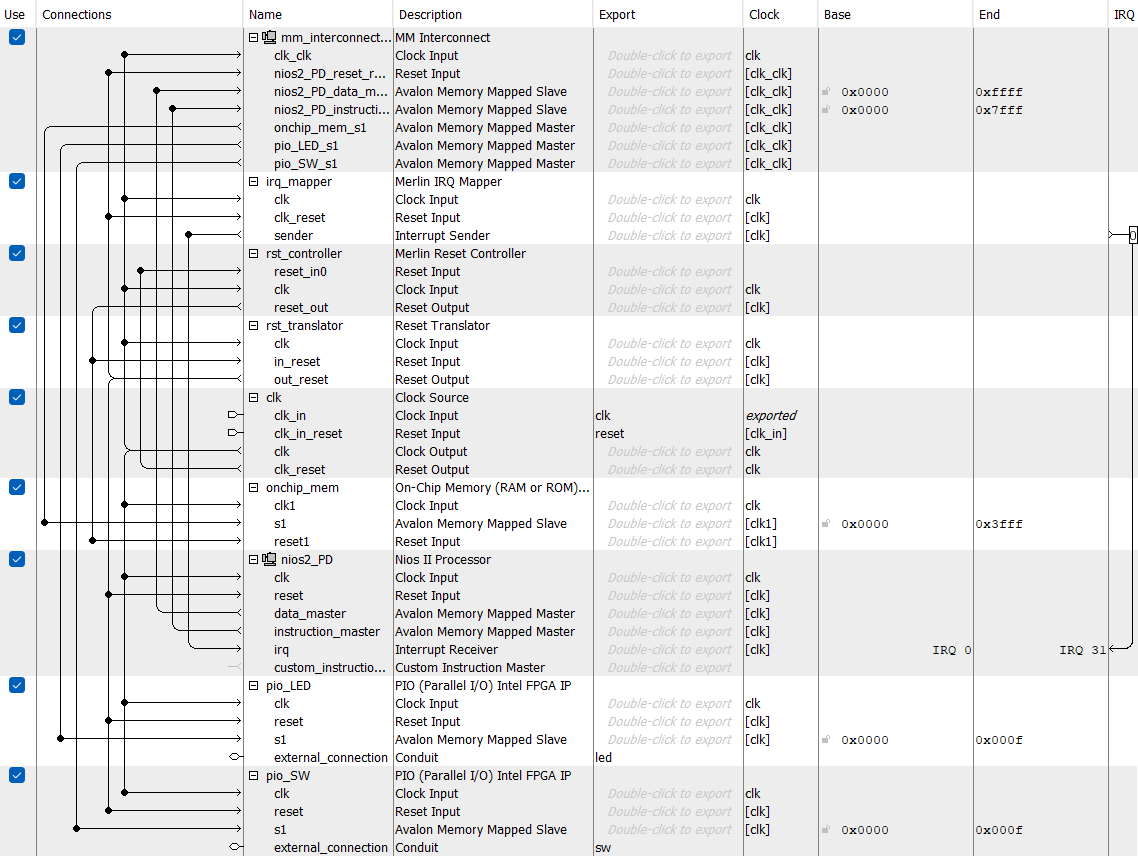


Рис.1-4. Show System with Platform Designer Interconnect

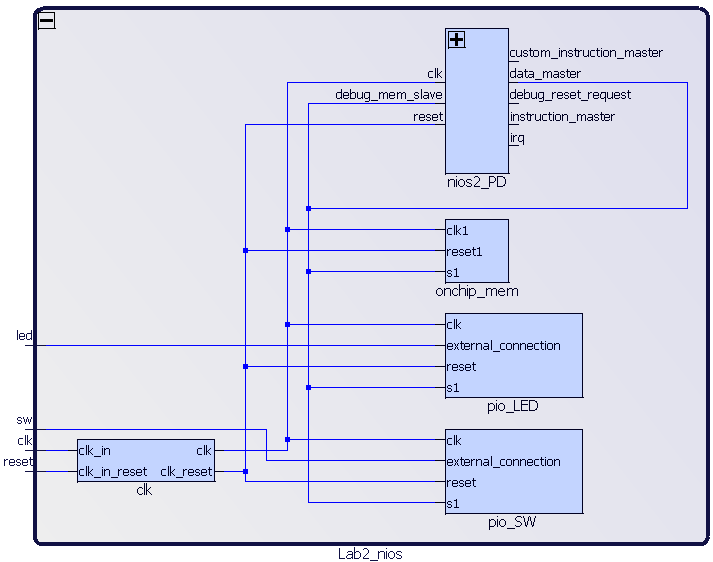


Рис.1-5. Schematic. In

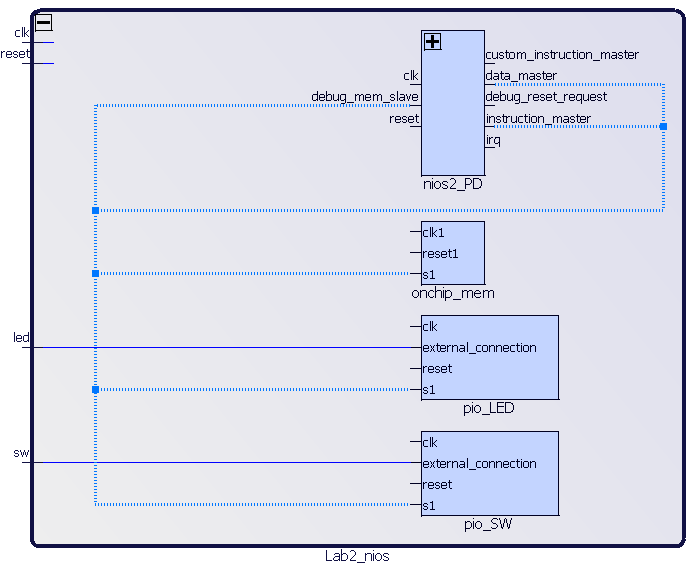


Рис.1-6. Schematic. clk

## Программирование Nios 2

Программа, написанная для soft-процессора Nios 2 представлена в листинге 1-3.

Листинг 1-3. Пример программы с использованием переменных из окружения.

|  |
| --- |
| **#include** "system.h"  **#include** "altera\_avalon\_pio\_regs.h"  **#include** <unistd.h>  **int** **main**(**void**) {  **int** sw;  **int** led = 0x00;  **int** count = 64;  **int** a = 0;  **while** (1)  {  **usleep** (300000);  **if**(sw != 0x00 && a == 0){  count = sw;  a = 1;  }  **if**(sw != 0x00 && sw < count)  count--;  **if**(sw ==0x00){  count = 0;  a = 0;  }  led = count;  }  } |

## Выводы

В результате были пройдены шаги из презентации для создания макета в Platform Designer и написания программы для soft-процессора Nios 2.