САНКТ-ПЕТЕРБУРГСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

ПЕТРА ВЕЛИКОГО

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

Институт компьютерных наук и технологий

Высшая школа интеллектуальных систем и суперкомпьютерных технологий

Отчет

по заданию lab 14

Дисциплина

«Технологии проектирования аппаратных средств компьютерных систем»

выполнил:

Курякин Д. А

группа:

преподаватель:

Антонов А. П.

Санкт-Петербург

2022

Оглавление

[1 Задание labn\_1 3](#_Toc105790972)

[1.1 Задание 3](#_Toc105790973)

[1.2 Создание макета в Platform Designer 3](#_Toc105790974)

[1.3 Анализ системы 4](#_Toc105790975)

[1.4 Программирование Nios 2 6](#_Toc105790976)

[1.5 Выводы 7](#_Toc105790977)

[2 Задание labn\_1s 7](#_Toc105790978)

[2.1 Задание 7](#_Toc105790979)

[2.2 Создание макета в Platform Designer 7](#_Toc105790980)

[2.3 Анализ системы 9](#_Toc105790981)

[2.4 Программирование Nios 2 11](#_Toc105790982)

[2.5 Выводы 12](#_Toc105790983)

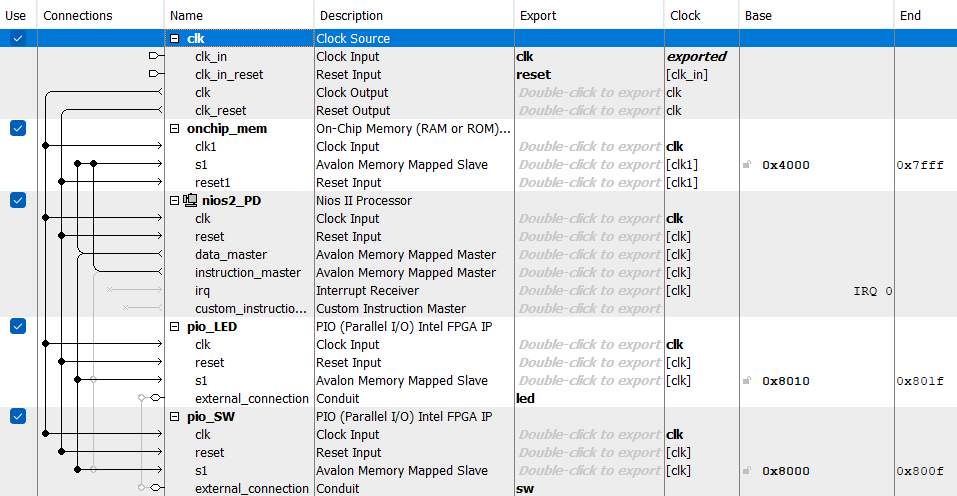
# Задание labn\_1

## 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. Lab1.sv

|  |
| --- |
| module Lab1(  input bit clk,  input bit sw,  input bit pbb,  output bit [7:0] led  );  Lab1\_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}]  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 {pbb}]  set\_input\_delay -add\_delay -clock [get\_clocks {clock}] 10.000 [get\_ports {sw}]  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.

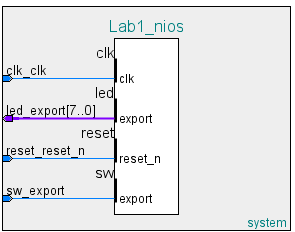


Рис.1-3. Block Symbol

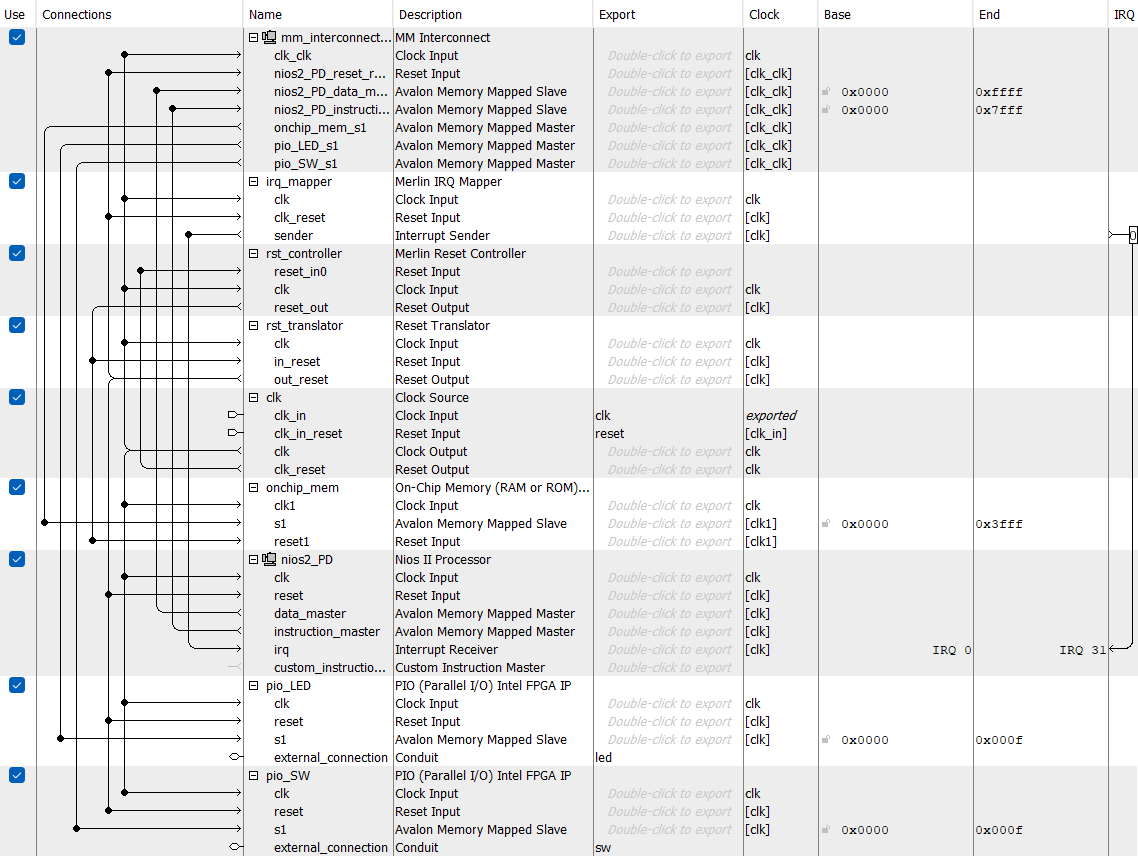


Рис.1-4. Show System with Platform Designer Interconnect

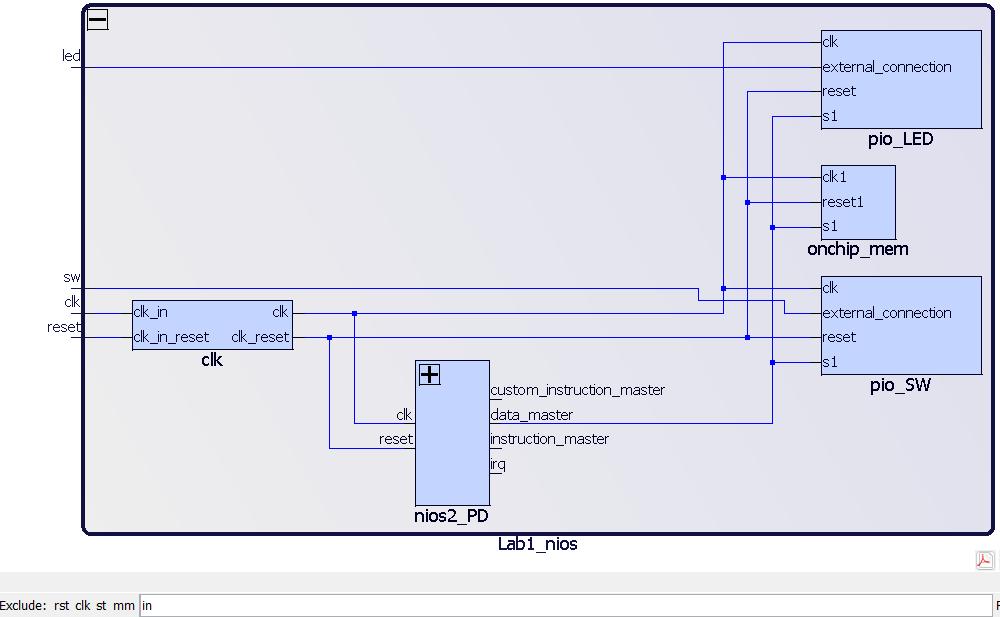


Рис.1-5. Schematic. In

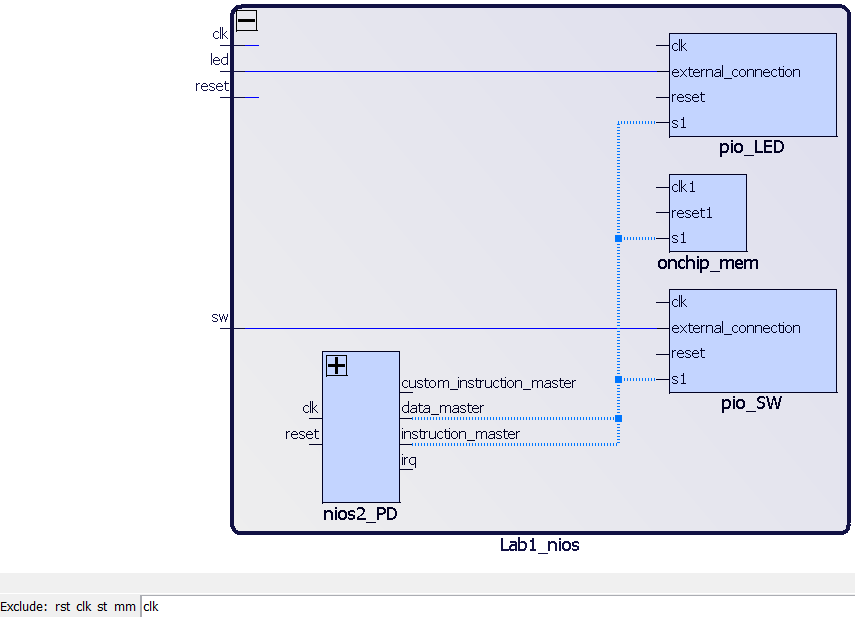


Рис.1-6. Schematic. clk

## Программирование Nios 2

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

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** 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;  } |

Листинг 1-4. Пример программы с использованием адресов.

|  |
| --- |
| **#include** "system.h"  **#include** "altera\_avalon\_pio\_regs.h"  **#include** <unistd.h>  **int** **main**(**void**)  {  **int** \*psw = (**int**\*) 0x8000;  **int** \*pled = (**int**\*) 0x8010;  **int** count = 64;  **while**( 1 )  {  **usleep** (500000);  **if** (\*psw == 0x1)  count++; /\* Continue 0-ff counting loop. \*/  **else**  count--; /\* Continue ff-0 counting loop. \*/  \*pled = ~count;  }  **return** 0;  } |

## Выводы

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

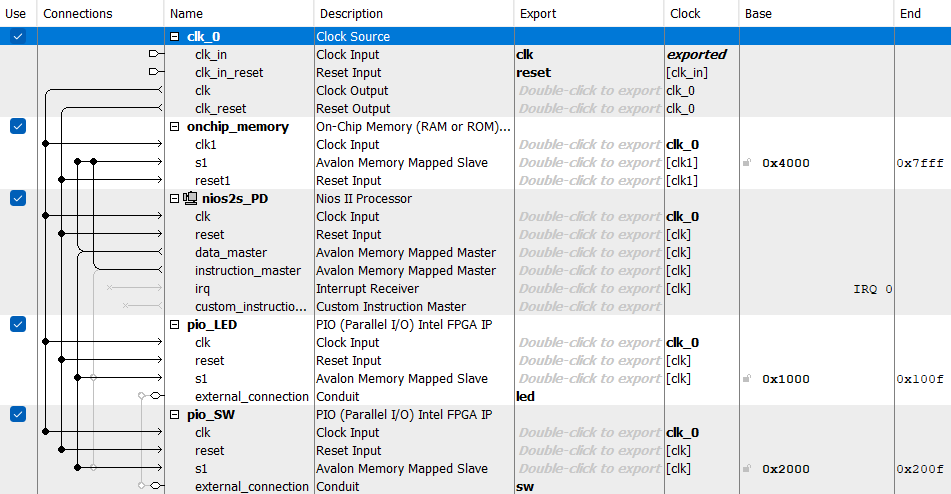
# Задание labn\_1s

## 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 Lab1s (  input wire clk,  output wire [7:0] led,  input wire reset,  input wire [7:0] sw  );  lab1s\_nios (  .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.

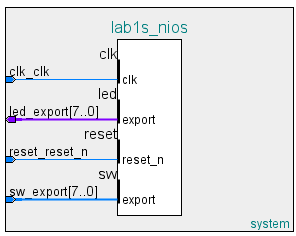


Рис.1-3. Block Symbol

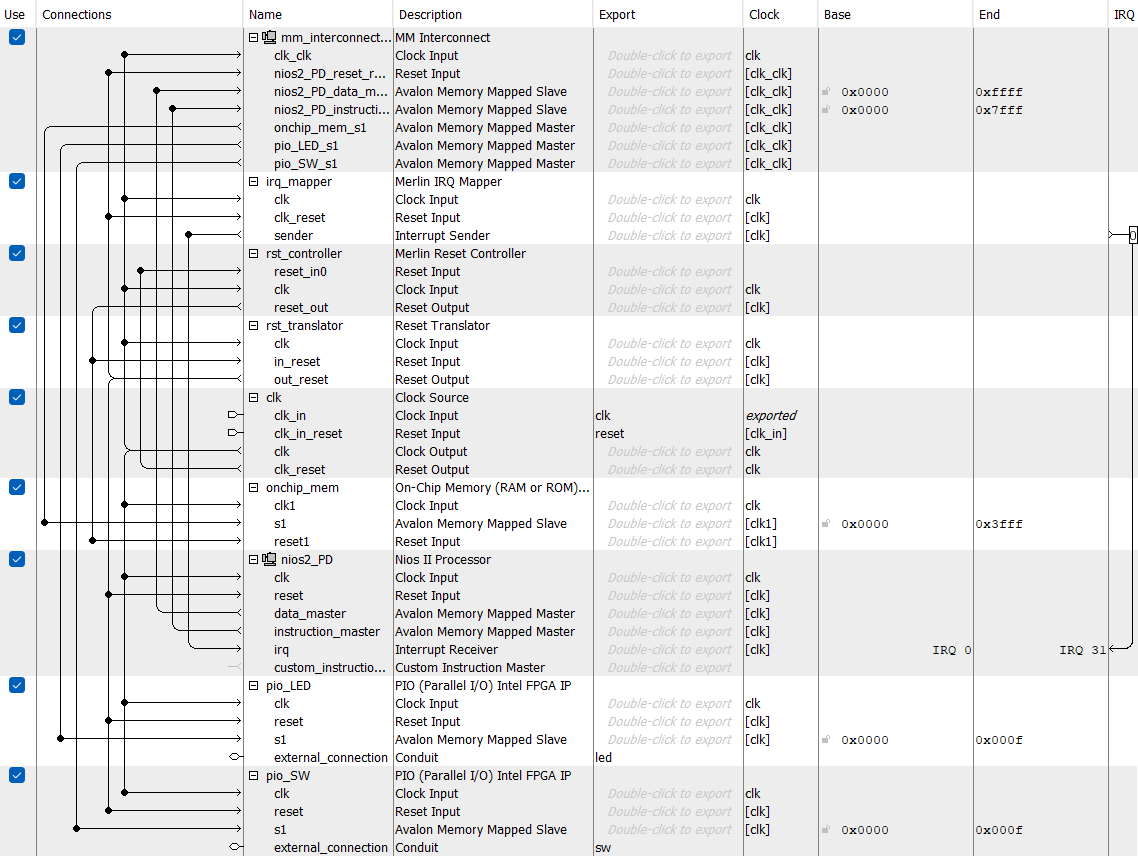


Рис.1-4. Show System with Platform Designer Interconnect

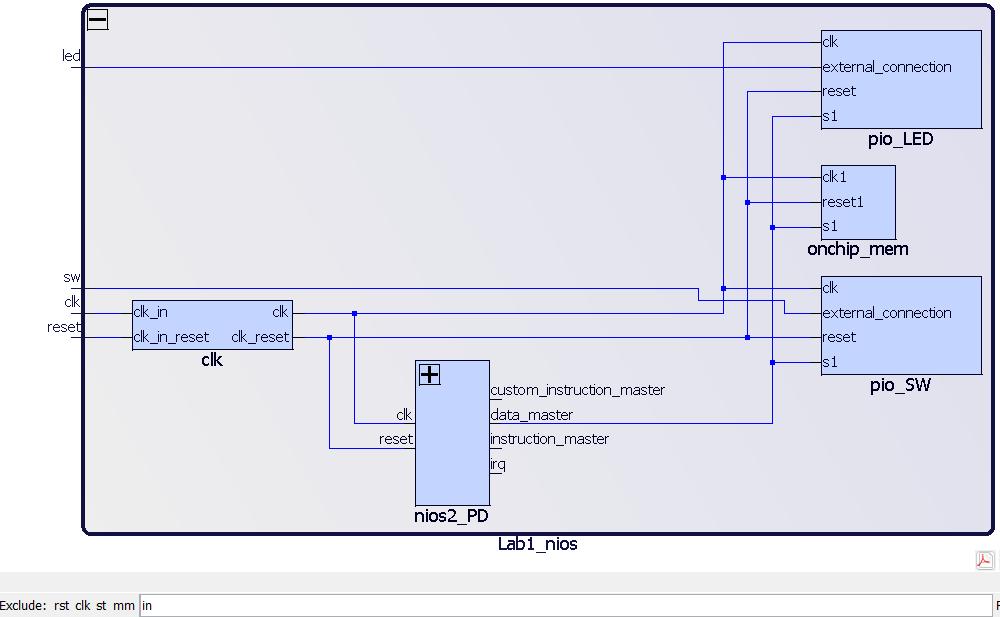


Рис.1-5. Schematic. In

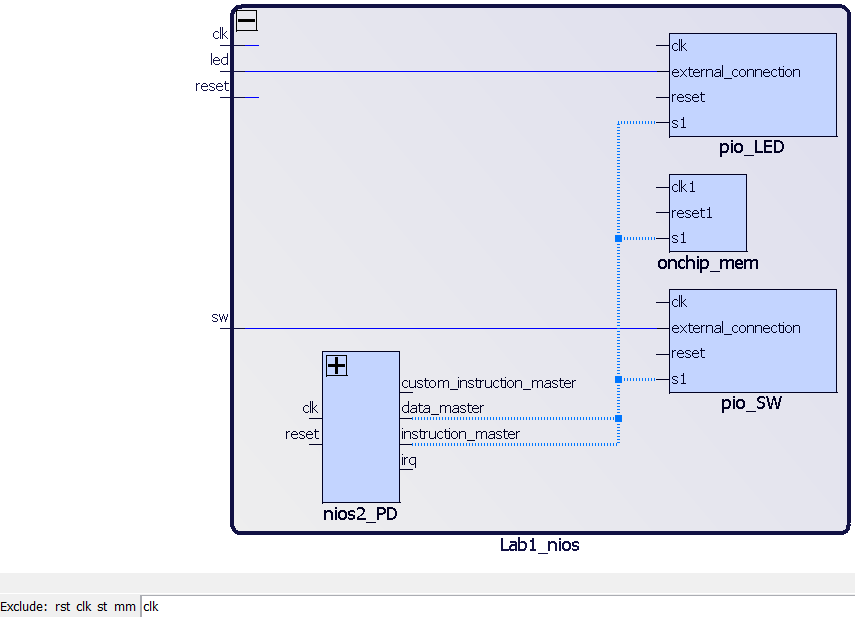


Рис.1-6. Schematic. clk

## 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** <unistd.h>  **int** **main**(**void**)  {  **char** \*psw = (**char**\*) 0x2000;  **char** \*pled = (**char**\*) 0x1000;  **char** count = 64;  **while** (1)  {  **usleep**(300000);  **if** (((\*psw)!=0x00) & (((\*psw)-1) > count))  count++;  **else**  count = 0;  \*pled = ~count;  }  **return** 0;  } |

## Выводы

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