МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Базовая кафедра «Вычислительные технологии»

ОТЧЕТ

по лабораторной работе №3
по дисциплине «Проектирование реконфигурируемых систем на кристалле»
Тема: Сравнение характеристик проектов на основе Nios II в кристалле Cyclone V при их аппаратной и программно-аппаратной реализации
Вариант 4

Студенты гр. 6309 _____ Васин А. М.
_____ Жвакин К. Э.
_____ Ладыженский Р. С.
Преподаватель _____ Шарагина Н.С.

Санкт-Петербург
2021

Цель работы.

Цель работы состоит в анализе особенностей проектирования и отладки и сравнении эффективности проектирования систем на базе soft-процессора Nios II и проектирования беспроцессорных аппаратных модулей при их реализации в условиях одной СнК.

Основные теоретические положения.

Один из способов оценки корректности проектов основан на моделировании, предполагающем отладку проекта на модели путем анализа реакций разрабатываемых схем на стимулирующие воздействия. Несмотря на высокую вероятность обнаружения имеющихся дефектов моделирование не всегда позволяет оценить работу схемы. Более эффективны методы, основанные на экспериментах с реальным оборудованием. Получившие в последнее время широкое распространение прототипные платы разработчика, содержащие ПЛИС, позволяют организовать подобные эксперименты.

Одно из основных достоинств СнК – гибкость решения различных задач. На одном и том же кристалле ПЛИС можно реализовать различные способы решения поставленной задачи, оценить результат и выбрать более подходящий в данном случае способ.

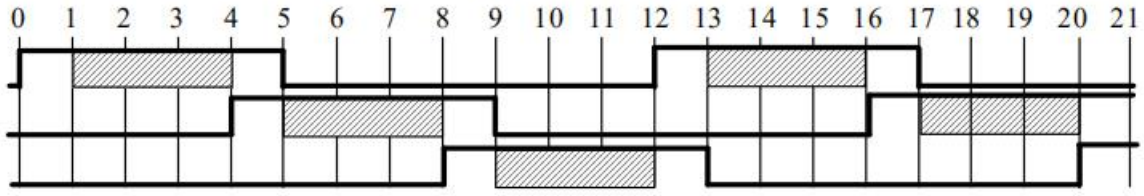


Рис. 1 - Диаграмма сигналов трехфазного управления

В работе предлагается реализовать два независимых решения задачи формирования сигналов трехфазного управления, аппаратно выполненные на одном кристалле ПЛИС фирмы «Altera».

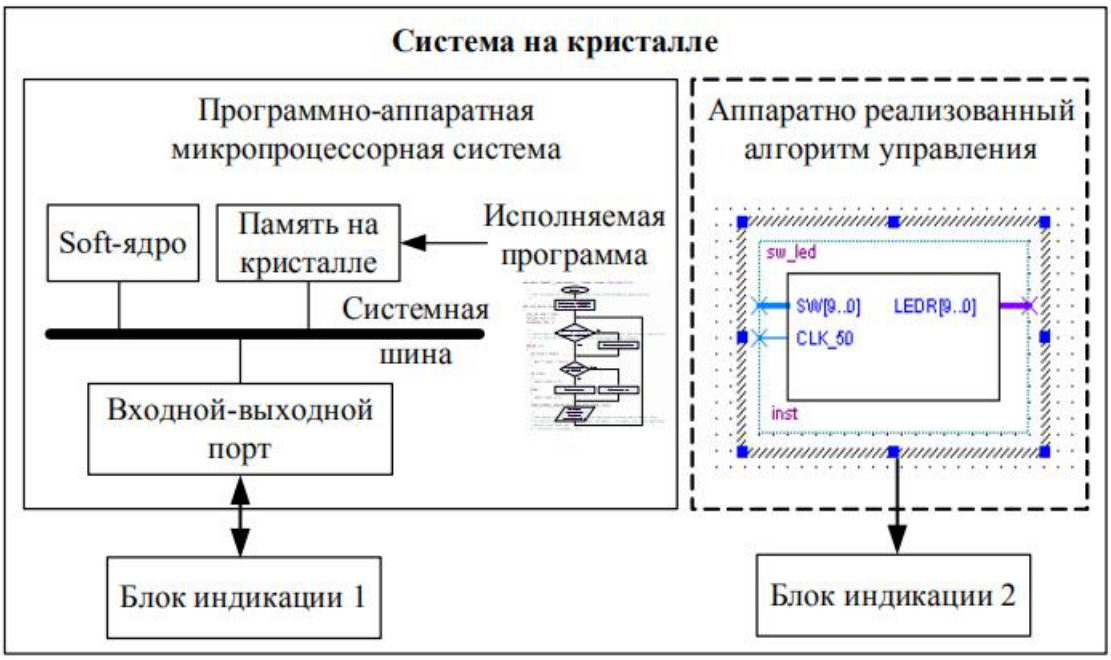


Рис. 2 - Устройство трехфазного управления

Первая система предполагает чисто аппаратную реализацию алгоритма, а вторая должна быть выполнена как программно-аппаратная микропроцессорная система на основе процессорного ядра Nios II. Пример временной диаграммы трехфазного управления с перекрытием на один такт приведен на рис. 1, а структурная схема устройства – на рис. 2.

Задание на работу.

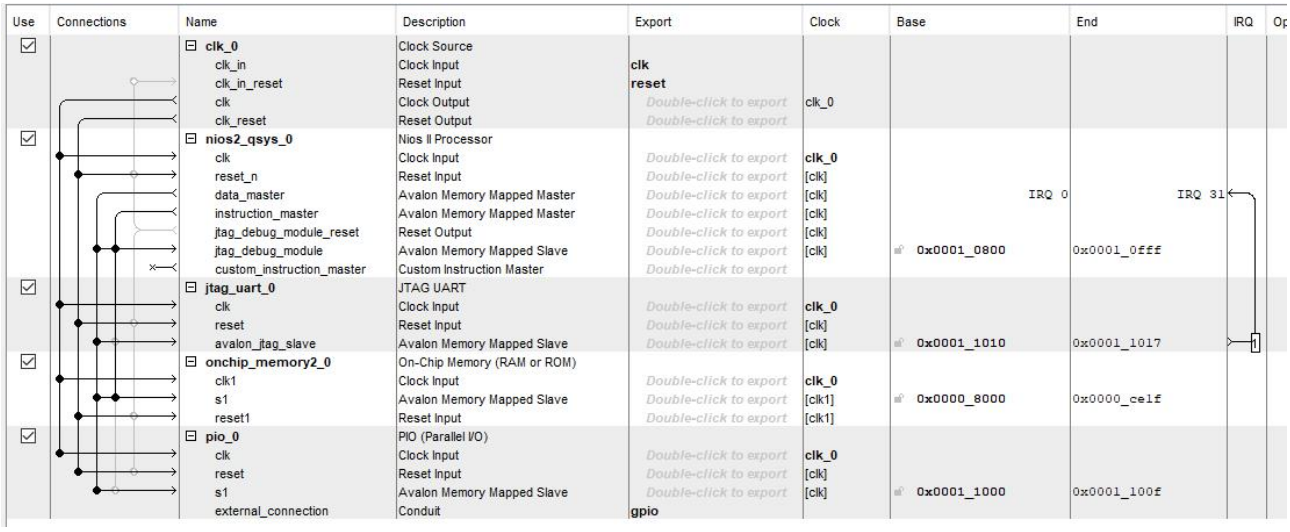
Разработать СнК с двумя независимыми решениями задачи трехфазного управления: аппаратным и программно-аппаратным. Выполнить имплементацию проекта СнК в ПЛИС отладочной платы DE0 фирмы «Terasic». Визуальный контроль работоспособности проекта осуществить путем подключения светодиодных индикаторов.

Выполнение работы.

Этап 1. Создание проекта в САПР Quartus II

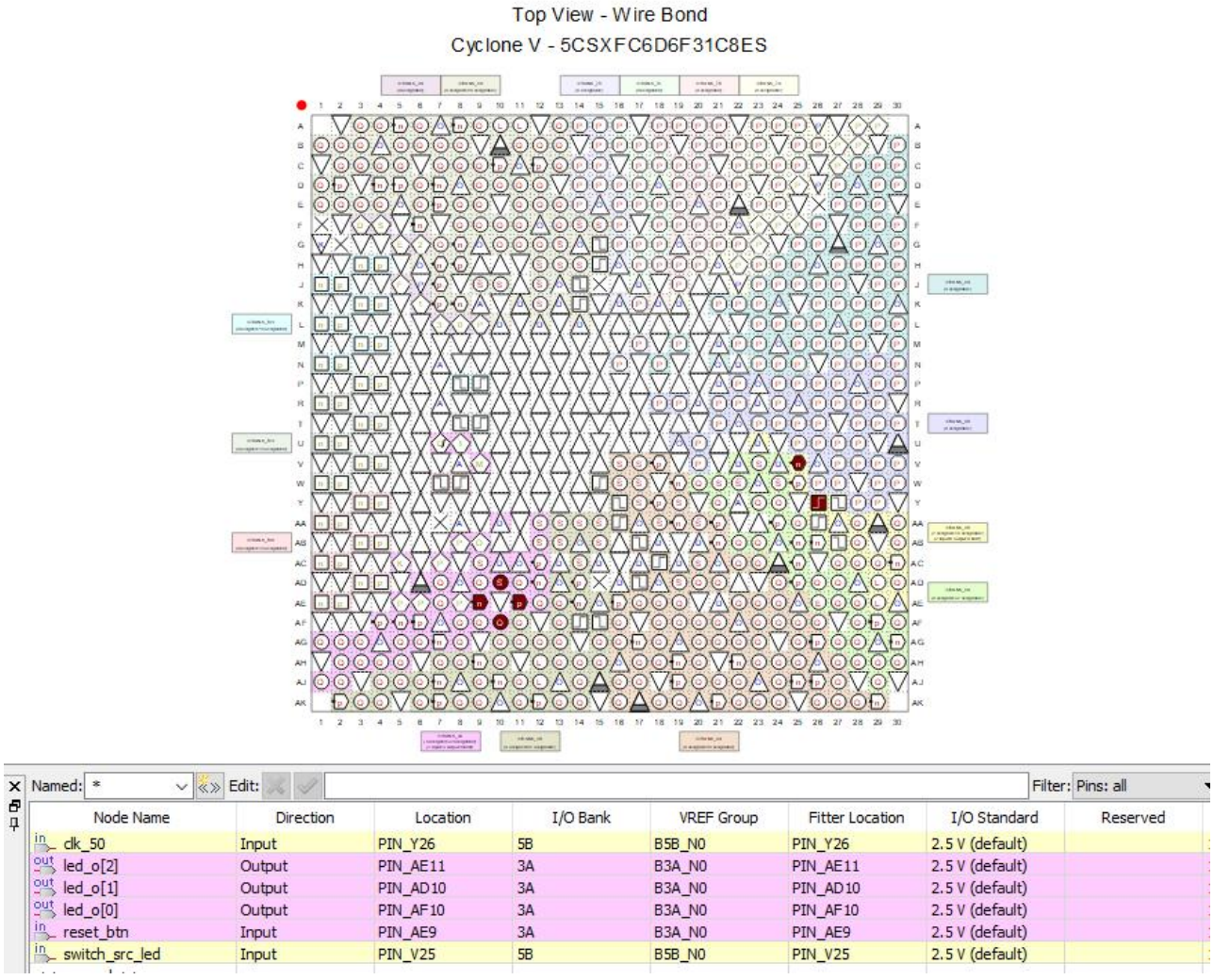
1. Выполним конфигурирование процессорной части СнК, а также сгенерируем файлы описания soft-процессора NIOS II со встроенной в кристалл

памятью программ и данных и блоком индикации. Выполним компиляцию проекта, назначим контакты, выполним загрузку процессорного ядра в ПЛИС.



Use	Connections	Name	Description	Export	Clock	Base	End	IRQ	Op
☑		⊟ clk_0	Clock Source						
		clk_in	Clock Input	clk					
		clk_in_reset	Reset Input	reset					
		clk	Clock Output	Double-click to export	clk_0				
		clk_reset	Reset Output	Double-click to export					
☑		⊟ nios2_qsys_0	Nios II Processor						
		clk	Clock Input	Double-click to export	clk_0				
		reset_n	Reset Input	Double-click to export	[clk]				
		data_master	Avalon Memory Mapped Master	Double-click to export	[clk]			IRQ 0	IRQ 31
		instruction_master	Avalon Memory Mapped Master	Double-click to export	[clk]				
		jtag_debug_module_reset	Reset Output	Double-click to export	[clk]				
		jtag_debug_module	Avalon Memory Mapped Slave	Double-click to export	[clk]	0x0001_0800	0x0001_0fff		
		custom_instruction_master	Custom Instruction Master	Double-click to export					
☑		⊟ jtag_uart_0	JTAG UART						
		clk	Clock Input	Double-click to export	clk_0				
		reset	Reset Input	Double-click to export	[clk]				
		avalon_jtag_slave	Avalon Memory Mapped Slave	Double-click to export	[clk]	0x0001_1010	0x0001_1017		
☑		⊟ onchip_memory2_0	On-Chip Memory (RAM or ROM)						
		clk1	Clock Input	Double-click to export	clk_0				
		s1	Avalon Memory Mapped Slave	Double-click to export	[clk1]	0x0000_8000	0x0000_ce1f		
		reset1	Reset Input	Double-click to export	[clk1]				
☑		⊟ pio_0	PIO (Parallel I/O)						
		clk	Clock Input	Double-click to export	clk_0				
		reset	Reset Input	Double-click to export	[clk]				
		s1	Avalon Memory Mapped Slave	Double-click to export	[clk]	0x0001_1000	0x0001_100f		
		external_connection	Conduit	gpio					

Рис. 3 - Результат конфигурирования системы



Node Name	Direction	Location	I/O Bank	VREF Group	Fitter Location	I/O Standard	Reserved
clk_50	Input	PIN_Y26	5B	B5B_N0	PIN_Y26	2.5 V (default)	
led_o[2]	Output	PIN_AE11	3A	B3A_N0	PIN_AE11	2.5 V (default)	
led_o[1]	Output	PIN_AD10	3A	B3A_N0	PIN_AD10	2.5 V (default)	
led_o[0]	Output	PIN_AF10	3A	B3A_N0	PIN_AF10	2.5 V (default)	
reset_btn	Input	PIN_AE9	3A	B3A_N0	PIN_AE9	2.5 V (default)	
switch_src_led	Input	PIN_V25	5B	B5B_N0	PIN_V25	2.5 V (default)	

Рис. 4 - Результат назначения контактов

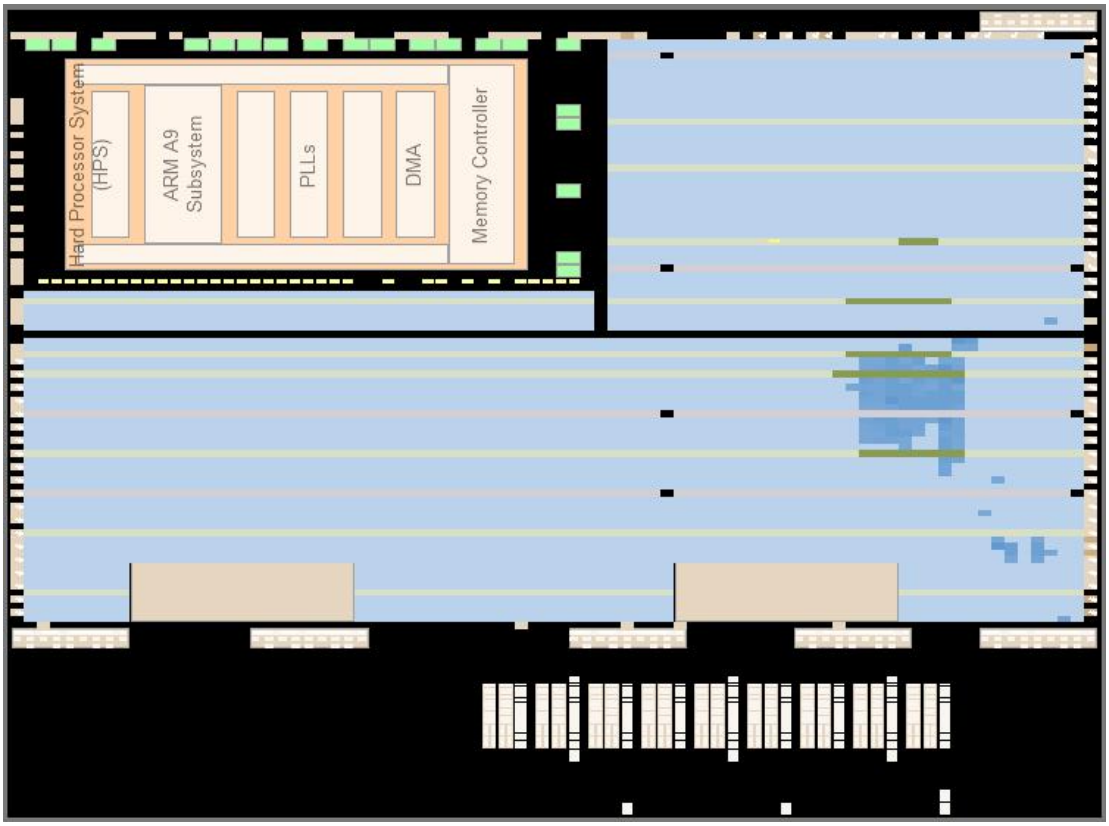


Рис. 5 - Расположение проекта в ПЛИС

2. Загрузим на выполнение программу трехфазного управления для Nios II. Листинг программы, соответствующий временной диаграмме на рис. 1, приведен в листинге 1. Работа программы построена на основе циклического формирования выходных сигналов. led[0]…led[2] – выходные значения сигналов управления для каждой фазы, Counter – счётчик тактов для фаз.

Листинг 1. Исходный код программы.

```
#include "sys/alt_stdio.h"
#include "system.h"
#include "altera_avalon_pio_regs.h"
#include <unistd.h>

#define FIRST_UP    24
#define FIRST_DOWN  10
#define SECOND_UP   8
#define SECOND_DOWN 18
#define THIRD_UP    16
#define THIRD_DOWN  2

int main()
{
  alt_putstr("Hello from Nios II!\n");

  int data_led = 0x00;
  alt_u8 led [3] = {0, };
```

```
  int counter = 0;
  /* Event loop never exits. */
  while (1)
  {
        counter++;
        switch(counter)
        {
        case FIRST_UP:     led[0] = 1; counter = 0; break;
        case FIRST_DOWN:   led[0] = 0; break;
        case SECOND_UP:    led[1] = 1; break;
        case SECOND_DOWN:  led[1] = 0; break;
        case THIRD_UP:     led[2] = 1; break;
        case THIRD_DOWN:   led[2] = 0; break;
        default: break;
        }

        data_led = led[2] << 2 | led[1] << 1 | led[0];
        IOWR_ALTERA_AVALON_PIO_DATA(PIO_0_BASE, data_led);

        usleep(50000);
  }

  return 0;
}
```

3. Оценка временных и аппаратных затрат на реализацию проекта представлена на рисунках ниже.

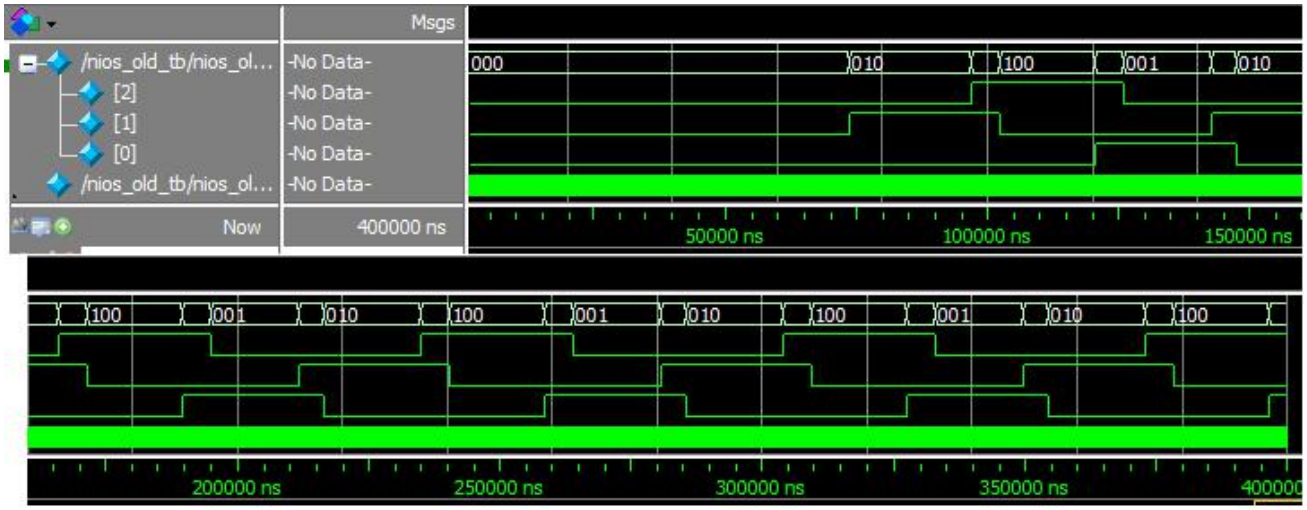


Рис. 6 - Результат моделирование проекта во временной области

Flow Summary	
Flow Status	Successful - Mon Dec 06 13:24:10 2021
Quartus II 64-Bit Version	13.0.1 Build 232 06/12/2013 SP 1 SJ Web Edition
Revision Name	lab_3
Top-level Entity Name	lab_3
Family	Cyclone V
Device	5CSXFC6D6F31C8ES
Timing Models	Preliminary
Logic utilization (in ALMs)	719 / 41,910 (2 %)
Total registers	933
Total pins	5 / 499 (1 %)
Total virtual pins	0
Total block memory bits	171,264 / 5,662,720 (3 %)
Total DSP Blocks	0 / 112 (0 %)
Total HSSI RX PCSs	0 / 9 (0 %)
Total HSSI PMA RX Deserializers	0 / 9 (0 %)
Total HSSI PMA RX ATT Deserializers	0
Total HSSI TX PCSs	0 / 9 (0 %)
Total HSSI PMA TX Serializers	0 / 9 (0 %)
Total HSSI PMA TX ATT Serializers	0 / 9 (0 %)
Total PLLs	0 / 15 (0 %)
Total DLLs	0 / 4 (0 %)

Рис. 7 - Оценка затрат на реализацию проекта

Рис. 8 - RTL - представление проекта

Этап 2. Аппаратная реализация алгоритма

В листинге 2 приведёно описание разработанного устройства, реализующего заданный алгоритм управления.

В данном проекте индикация светодиодов осуществляется от выбираемого источника. Если переменная switch_src_led равна единице, то светодиоды будут отображать результат работы soft-процессора Nios II, в противном случае светодиоды будут отображать результат реализованного на этом этапе аппаратного решения.

```
`define FIRST_UP    5'd24
`define FIRST_DOWN  5'd10
`define SECOND_UP   5'd8
`define SECOND_DOWN 5'd18
`define THIRD_UP    5'd16
`define THIRD_DOWN  5'd2
```

```
module lab_3 (
       input logic clk_50,
       input logic reset_btn,
       input logic switch_src_led,
       output [2:0] led_o
       );

       logic [2:0] led_nios;
       logic [2:0] led_fpga;

       assign led_o = switch_src_led ? led_nios : led_fpga;

nios_old nios_instance (
             .clk_clk (clk_50),       //   clk.clk
             .gpio_export (led_nios),   //  gpio.export
             .reset_reset_n (reset_btn) // reset.reset_n
       );


logic [4:0] counter;

logic clk_10Hz;
generator #(500_0000) //for 10Hz is 500_0000
    generator_100Hz(
       .nreset_i (reset_btn),
       .i_clk (clk_50),
       .o_clk (clk_10Hz)
       );

always @(posedge clk_10Hz, negedge reset_btn)
begin
    if(!reset_btn)
    begin
        led_fpga = 3'b000;
        counter = 4'd0;
    end
    else
    begin
        counter++;
        case (counter)
            `FIRST_UP:begin led_fpga[0] = 1'b1; counter = 5'd0; end
            `FIRST_DOWN:    led_fpga[0] = 1'b0;
            `SECOND_UP:     led_fpga[1] = 1'b1;
            `SECOND_DOWN:   led_fpga[1] = 1'b0;
            `THIRD_DOWN:    led_fpga[2] = 1'b0;
            `THIRD_UP:      led_fpga[2] = 1'b1;
           default: ;//begin led_fpga[0] = led_fpga0]; led_fpga[1] = led_fpga[1];
led_fpga[2] = led_fpga[2]; end
        endcase
    end
end

endmodule
```

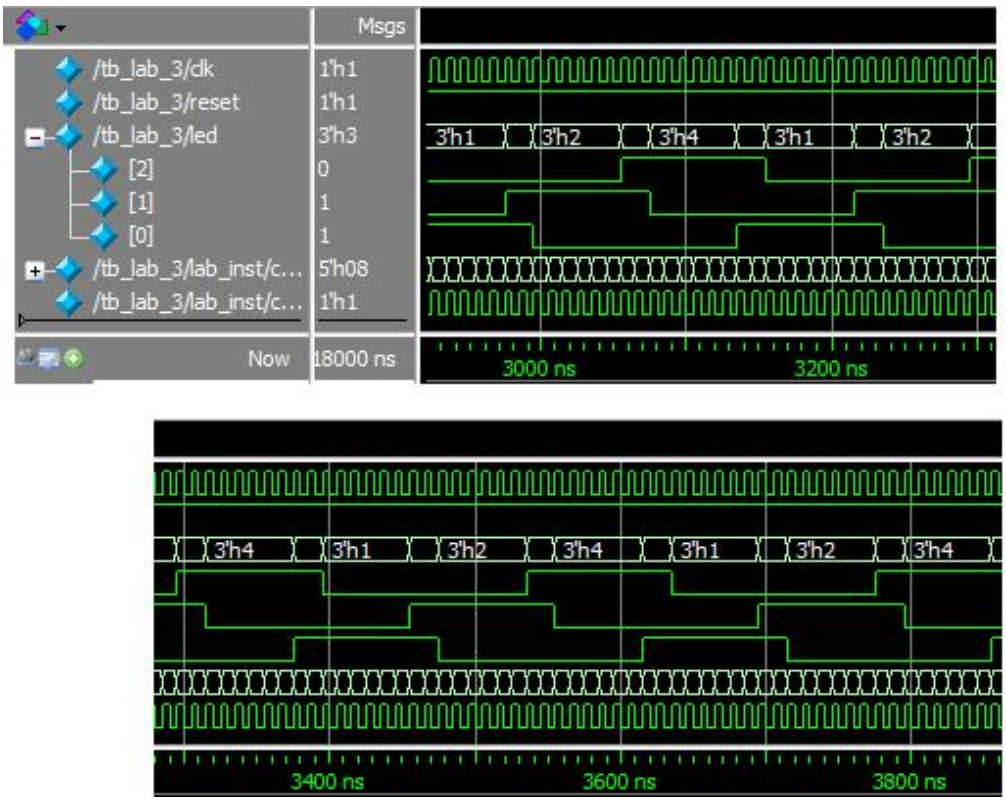


Рис. 9 - Результат моделирования проекта во временной области

Flow Status	Successful - Mon Dec 06 13:45:26 2021
Quartus II 64-Bit Version	13.0.1 Build 232 06/12/2013 SP 1 SJ Web Edition
Revision Name	lab_3
Top-level Entity Name	lab_3
Family	Cyclone V
Device	5CSXFC6D6F31C8ES
Timing Models	Preliminary
Logic utilization (in ALMs)	27 / 41,910 (< 1 %)
Total registers	46
Total pins	5 / 499 (1 %)
Total virtual pins	0
Total block memory bits	0 / 5,662,720 (0 %)
Total DSP Blocks	0 / 112 (0 %)
Total HSSI RX PCSs	0 / 9 (0 %)
Total HSSI PMA RX Deserializers	0 / 9 (0 %)
Total HSSI PMA RX ATT Deserializers	0
Total HSSI TX PCSs	0 / 9 (0 %)
Total HSSI PMA TX Serializers	0 / 9 (0 %)
Total HSSI PMA TX ATT Serializers	0 / 9 (0 %)
Total PLLs	0 / 15 (0 %)
Total DLLs	0 / 4 (0 %)

Рис. 10 - Оценка затрат на компиляцию проекта

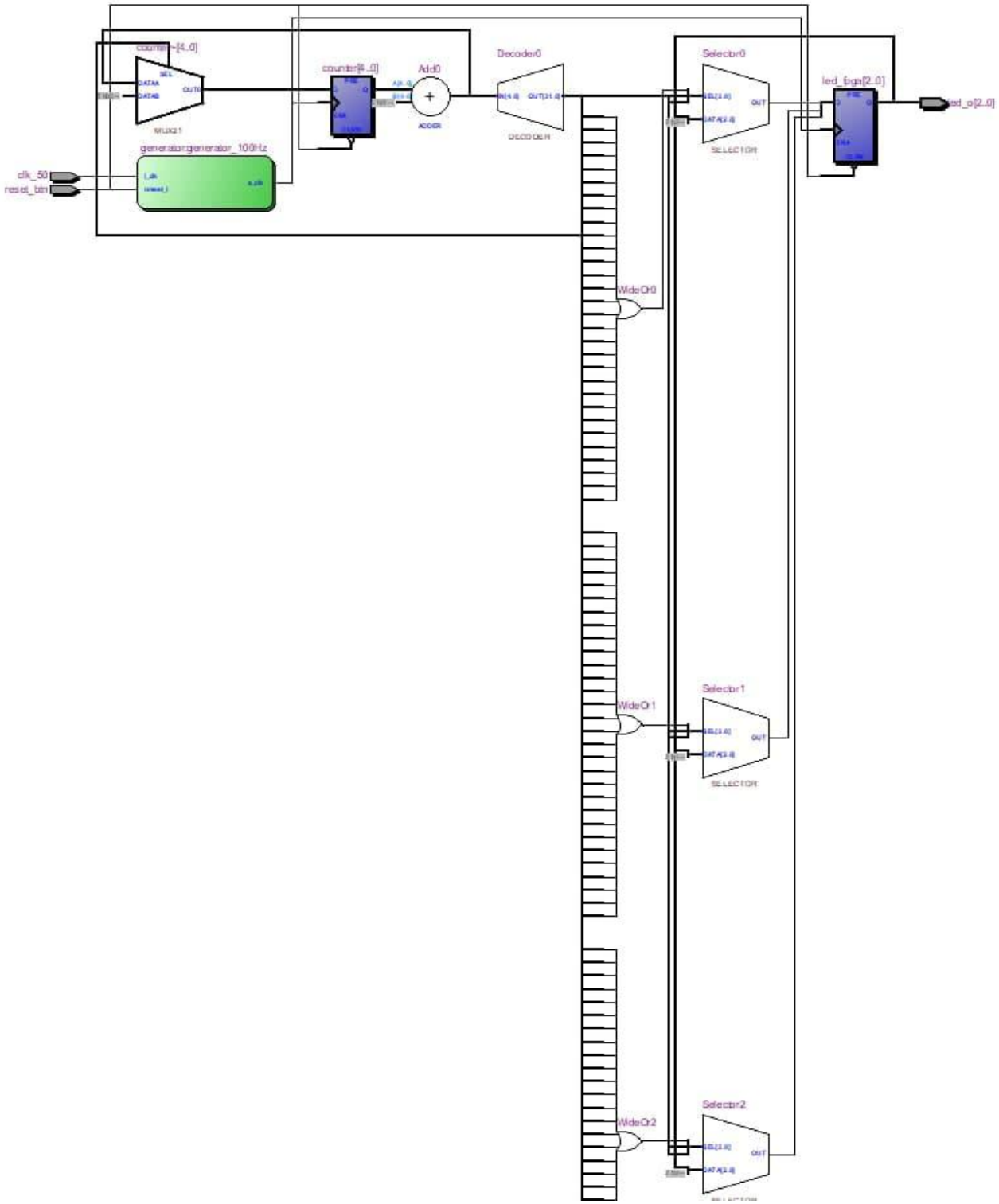


Рис. 11 - RTL-представление проекта

Выводы.

В ходе выполнения данной лабораторной работы был произведён анализ особенностей проектирования и отладки систем на базе soft-процессора Nios II и беспроцессорного аппаратного модуля при их реализации в условиях одной СнК.

Сравнения проводились с отключенными задержками на аппаратном входном тактовом сигнале 50МГц, минимальные периоды составили 69000 нс и 241 нс соответственно. Таким образом, разница составила 286 раз по скорости работы в пользу аппаратной реализации.

Далее, сравнивая затраты на реализацию проекта, можно отметить, что применение аппаратных модулей позволяет уменьшить использование логики в 26 раз, сократить количество используемых регистров в 20 раз, а также избавиться от использования блочной памяти ПЛИС (в которой хранится исполняемый код программы). Всё это обусловленно отсутсвием необходимости синтеза процессорного ядра для выполнения одной и той же задачи.

Подводя итоги, можно отметить эффективность применения аппаратных модулей для выполнения задач, требующих большого быстродействия, в то время как применение soft-процессоров позволяет снизить временные затраты на выполнение проекта.