|  |  |
| --- | --- |
| *voenmeh* | МИНОБРНАУКИ РОССИИ  федеральное государственное бюджетное образовательное учреждение  высшего образования  **«Балтийский государственный технический университет «ВОЕНМЕХ» им. Д.Ф. Устинова»**  **(БГТУ «ВОЕНМЕХ» им. Д.Ф. Устинова»)** |
| БГТУ.СМК-Ф-4.2-К5-01 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Факультет |  | И |  | Информационные и управляющие системы |
|  |  | шифр |  | наименование |
| Кафедра |  | И4 |  | Радиоэлектронные системы управления |
|  |  | шифр |  | наименование |
| Дисциплина |  | Проектирование цифровых систем на ПЛИС | | |

КУРСОВАЯ РАБОТА

на тему

|  |
| --- |
|  |
| **«Электронный влагомер»** |
|  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Выполнил студент группы | | | |  | И4М31 |
| Густов В. В. | | | | | |
| Фамилия И.О. | | | | | |
| **РУКОВОДИТЕЛЬ** | | | | | |
| Ширшов А. Д. | |  |  | | |
| Фамилия И.О. Подпись | | | | | |
| Оценка |  | | | |  |
| «\_\_\_\_\_» |  | | | | 2019 г. |

САНКТ-ПЕТЕРБУРГ

2019 г.

**РЕФЕРАТ**

Пояснительная записка курсовой работы содержит: 17 страниц, 6 рисунков, 5 источников информации.

ПЛИС, ДАТЧИК ВЛАЖНОСТИ, ДЕЛИТЕЛЬ ЧАСТОТЫ, СЕМИСЕГМЕНТНЫЙ ИНДИКАТОР, ЯЗЫК *VHDL*

Курсовая работа направлена на проектирование электронного влагомера на ПЛИС.

В ходе работы было выполнено следующие:

* Разработан алгоритм работы устройства.
* Разработана функциональная схема электронного влагомера.
* Смоделирована работа влагомера в среде *Quartus*.
* Произведена отладка на макете *DE1-SoC*.

**СОДЕРЖАНИЕ**

ОПРЕДЕЛЕНИЯ, ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ……………...4

ВВЕДЕНИЕ………………………………………………………………5

1. АНАЛИЗ ЗАДАЧИ…………………………………………………...6
2. РАЗРАБОТКА ОПИСАНИЯ УСТРОЙСТВА……………………...8
   1. ДЕЛИТЕЛЬ ЧАСТОТЫ………………………………………….8
   2. СЧЕТЧИК…………………………………………………………9
   3. УЗЕЛ РАСЧЕТА ЗНАЧЕНИЯ ВЛАЖНОСТИ……………......10
   4. ДЕШИФРАТОР ДЛЯ СИ……………………………………….12
   5. СХЕМА ВЛАГОМЕРА………………………………………….13
3. ОТЛАДКА НА ПЛАТЕ……………………………………………...15

ЗАКЛЮЧЕНИЕ……………………………………………………….....16

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ…………………...17

# **ОПРЕДЕЛЕНИЯ, ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ**

ПЛИС – программируемая логическая интегральная схема*.*

*VHDL – VHSIC (Very high speed integrated circuits) Hardware Description Language.*

СИ – семисегментный индикатор.

# **ВВЕДЕНИЕ**

Регулирование и автоматизация многих промышленных процессов требует точного и достоверного измерения влажности. Управляемые ПЛИС датчики влажности представляет собой универсальное решение для измерения влажности. Благодаря простоте обслуживания, обширному набору функций и возможности расширения, эти датчики доказывают свою надежность в различных технологических процессах. Данные приборы используются в промышленных процессах сушки, системах контроля и управления, климатических установках стерильных и складских помещений, лабораториях и др.

Приборы могут оснащаться CИ, предназначенным для отображения измеряемых значений. Используемые датчики отличаются повышенной точностью, надежностью и стабильностью.

Задачей данной курсовой работы является разработка электронного влагомера на ПЛИС (*Altera Cyclone V 5CSEMA5F31C6N, DE1-SoC*) для вывода текущей относительной влажности на СИ.

Работа состоит из трех частей, в первой части приводится и описывается структурная схема влагомера. Во второй части описываются алгоритмы работы узлов на языке описания аппаратуры, приводятся результаты моделирования разработанных алгоритмов и общая схема влагомера. В третьей части производится отладка разработанной схемы на плате (*DE1-SoC*).

# **1 АНАЛИЗ ЗАДАЧИ**

Произведем анализ задачи.

Проектируемый прибор должен получать значение с датчика влажности, производить расчет и выводить на пять СИ значение влажности.

Проектируемый прибор должен работать автономно, и обновлять информацию с датчиков по истечению определенного промежутка времени.

Структурная схема влагомера представлена на рисунке 1.

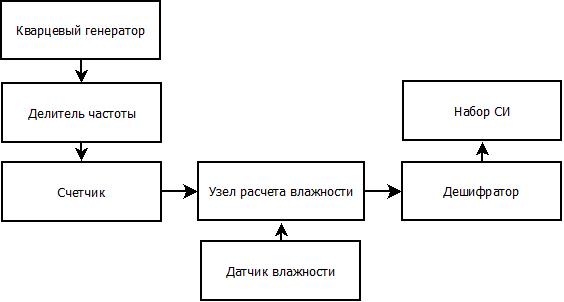


Рисунок 1 – Структурная схема влагомера.

Для реализации влагомера потребуется блок «Кварцевый генератор», который предназначен для получения фиксированной частоты с высокой температурной и временной стабильностью, низким уровнем фазовых шумов. На выходе блока «Кварцевый генератор» формируются импульсы, частота которых много больше одной секунды, для получения секунд предназначен блок «Делитель частоты». Выходом блока делителя частоты являются прямоугольные сигналы с периодом 1 секунда. Точность периода зависит от точности кварцевого генератора.

Сигналы от блока «Делитель частоты» поступают на счетчик, который накапливает сигналы до 10 (время обновления значений влажности). При накоплении 10 сигналов, на выходе блока «Счетчик» выводится логическая единица, банк суммы обнуляется, выход обнуляется и цикл продолжается. Логическая единица на выходе блока «Счетчик» запускает работу блока «Узел расчета влажности», который обрабатывает сигналы с блока «Датчик влажности», и выдает выходной сигнал на блок «Дешифратор», который в свою очередь представляет данные в виде, понятные для передачи СИ.

Передача обработанных данных для блока «Набор СИ» обеспечивает отображение значения на СИ.

# **2 РАЗРАБОТКА ОПИСАНИЯ УСТРОЙСТВА**

Рассмотрим каждый блок на структурной схеме (рисунок 1) как «черный ящик», с некоторыми входными и выходными параметрами [2]. Поведение структурных компонентов были разработаны на языке описания аппаратуры *VHDL* [1,3] в среде разработки *Quartus* [4].

## **2.1 ДЕЛИТЕЛЬ ЧАСТОТЫ**

Делитель частоты переключается через фиксированное число тактов. Ниже представлено описание поведения делителя частоты с 50МГц до 1 Гц.

*entity Div\_50Mhz\_to\_1Hz is*

*port( clk:in std\_logic; clk\_out:out std\_logic);*

*end Div\_50Mhz\_to\_1Hz;*

*architecture div\_behavior of Div\_50Mhz\_to\_1Hz is*

*begin*

*process(clk)*

*variable cnt : integer range 0 to 50000000;*

*begin*

*if(clk'event and clk = '1')*

*then*

*if(cnt >= 25000000)*

*then*

*clk\_out <= '1';*

*else*

*clk\_out <= '0';*

*end if;*

*if(cnt = 50000000)*

*then*

*cnt := 0;*

*else*

*cnt := cnt + 1;*

*end if;*

*end if;*

*end process;*

*end div\_behavior;*

На рисунке 2 представлен результат моделирования работы делителя частоты на десять (для наглядности количество тактов было сокращено в 1000000 раз).

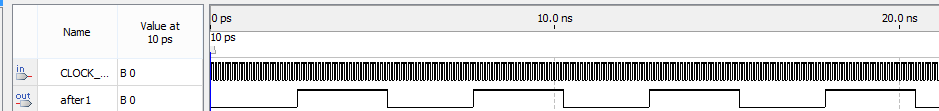
****

Рисунок 2 – Моделирование работы делителя частоты.

## **2.2 СЧЕТЧИК**

Значения счетчика увеличивается по каждому тактовому сигналу на входе, приходящему с блока «Кварцевый генератор». Счетчик обнуляется при 10 тактах, и при обнулении выводит логическую единицу на вывод блока «Счетчик». Ниже представлен фрагмент кода счетчика на языке VHDL [5].

*entity count is*

*port( clk:in std\_logic; c\_out:out std\_logic);*

*end count;*

*architecture count\_behavior of count is*

*begin*

*process(clk)*

*variable cnt : integer range 0 to 10;*

*begin*

*if(clk'event and clk = '1')*

*then*

*if(cnt = 10)*

*then*

*cnt := 0;*

*c\_out <= '1';*

*else*

*cnt := cnt + 1;*

*c\_out <= '0';*

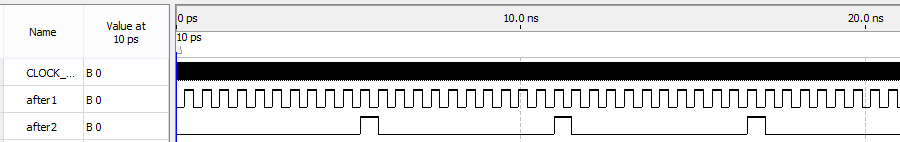
*end if;*

*end if;*

*end process;*

*end count\_behavior;*

На рисунке 3 представлен результат моделирования счетчика (для наглядности количество тактов было сокращено в 1000000 раз).

****Рисунок 3 – Моделирование счетчика.

## **2.3 УЗЕЛ РАСЧЕТА ЗНАЧЕНИЯ ВЛАЖНОСТИ**

На вход блока «Узел расчета значения влажности» подается разрешающий сигнал (от блока «Счетчик») и сигнал с данными с датчика влажности.

Ввиду того, что для программирования на плате требуется реальный датчик влажности, которого не имеется в наличии, он будет заменен эквивалентной схемой. В качестве замены будет использован набор переключателей, который будет имитировать двоичный код, являющийся представлением показаний, получаемые с датчика влажности. Имея 10 переключателей можно имитировать значение от 0 до 1023.

При подаче разрешающего сигнала блок «Узел расчета значения влажности», считываются значения с выхода блока «Датчик влажности», который в нашем случае представляется двоичным кодом числа. Данные обрабатываются и переводятся в позиционный код, который в дальнейшем нормируется до значений от 0 до 100%.

После нормирования, число разбивается на 5 цифр (три цифры до запятой, 2 цифры после запятой), преобразуются в *BCD*-код и передается в блок «Дешифратор». Ниже представлен код для блока «Узел расчета значения влажности».

*entity refresher is*

*port(*

*clock:in std\_logic;*

*input:in std\_logic\_vector(0 to 9);*

*to\_seg4:out std\_logic\_vector(3 downto 0);*

*to\_seg3:out std\_logic\_vector(3 downto 0);*

*to\_seg2:out std\_logic\_vector(3 downto 0);*

*to\_seg1:out std\_logic\_vector(3 downto 0);*

*to\_seg0:out std\_logic\_vector(3 downto 0)*

*);*

*end refresher;*

*architecture refresher\_behavior of refresher is*

*begin*

*process(clock)*

*variable sum : integer range 0 to 1023;*

*variable s4 : integer range 0 to 9;*

*variable s3 : integer range 0 to 9;*

*variable s2 : integer range 0 to 9;*

*variable s1 : integer range 0 to 9;*

*variable s0 : integer range 0 to 9;*

*variable temp : integer range 0 to 100;*

*begin*

*if(clock'event and clock = '1') then*

*sum:= 0;*

*if (input(9) = '1') then sum:=sum+512; end if;*

*if (input(8) = '1') then sum:=sum+256; end if;*

*if (input(7) = '1') then sum:=sum+128; end if;*

*if (input(6) = '1') then sum:=sum+64; end if;*

*if (input(5) = '1') then sum:=sum+32; end if;*

*if (input(4) = '1') then sum:=sum+16; end if;*

*if (input(3) = '1') then sum:=sum+8; end if;*

*if (input(2) = '1') then sum:=sum+4; end if;*

*if (input(1) = '1') then sum:=sum+2; end if;*

*if (input(0) = '1') then sum:=sum+1; end if;*

*temp:= sum\*100 / 1023;*

*if (integer(temp/10) > 9)*

*then*

*s4:=1; s3:=0; s2:=0;*

*else*

*s4:=0; s3:=temp / 10 ; s2:=temp rem 10;*

*end if;*

*sum:=sum\*100-1023\*temp;*

*temp:=sum\*100/1023;*

*s1:=temp / 10;*

*s0:=temp rem 10;*

*to\_seg4 <= CONV\_STD\_LOGIC\_VECTOR(s4, 4);*

*to\_seg3 <= CONV\_STD\_LOGIC\_VECTOR(s3, 4);*

*to\_seg2 <= CONV\_STD\_LOGIC\_VECTOR(s2, 4);*

*to\_seg1 <= CONV\_STD\_LOGIC\_VECTOR(s1, 4);*

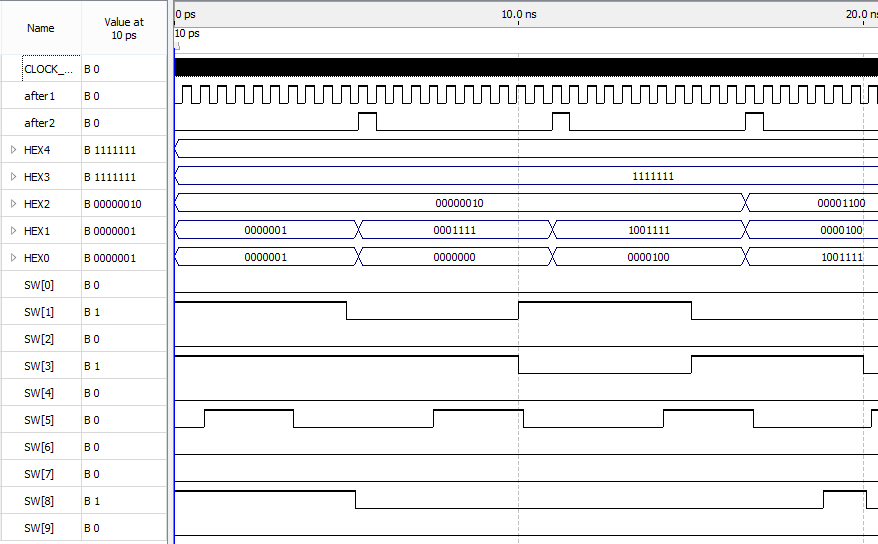
*to\_seg0 <= CONV\_STD\_LOGIC\_VECTOR(s0, 4);*

*end if;*

*end process;*

*end refresher\_behavior;*

На рисунке 4 представлен результат моделирования узла обработки. Как видно из рисунка, процесс обновления выходных сигналов *HEX* происходит только при наличии логической единицы на входе.

Рисунок 4 – Моделирование блока «Узел обработки».

## **2.4 ДЕШИФРАТОР ДЛЯ СИ**

На вход дешифратора подается BCD код числа, который преобразуется в вектор из 7 элементов. Каждому из 7 элементов соответствует сегменту индикатора в СИ. Ниже представлен код для блока «Дешифратор».

*entity BCD\_to\_7seg is*

*port(*

*BCD:in std\_logic\_vector(3 downto 0);*

*seg:out std\_logic\_vector(0 to 6)*

*);*

*end BCD\_to\_7seg;*

*architecture conv\_behavior of BCD\_to\_7seg is*

*begin*

*process(BCD)*

*begin*

*if BCD = "0000" then seg <= "0000001";--0*

*elsif BCD = "0001" then seg <= "1001111";--1*

*elsif BCD = "0010" then seg <= "0010010";--2*

*elsif BCD = "0011" then seg <= "0000110";--3*

*elsif BCD = "0100" then seg <= "1001100";--4*

*elsif BCD = "0101" then seg <= "0100100";--5*

*elsif BCD = "0110" then seg <= "0100000";--6*

*elsif BCD = "0111" then seg <= "0001111";--7*

*elsif BCD = "1000" then seg <= "0000000";--8*

*elsif BCD = "1001" then seg <= "0000100";--9*

*else seg <= "1001001";--err*

*end if;*

*end process;*

*end conv\_behavior;*

## **2.5 ФУНКИЦОНАЛЬНАЯ СХЕМА**

Итоговая функциональная схема влагомера (рисунок 5) содержит:

* Кварцевый генератор (*CLOCK\_50[0*]).
* Делитель частоты (*DIV\_50MHZ\_TO\_1HZ*).
* Счетчик (*COUNT*).
* Узел обработки (*REFRESHER*).
* Датчик влажности (эквивалентная схема, *SW* [0-9]).
* Дешифраторы (*BCD\_TO\_7\_SEG*).
* СИ (*HEX* [0..4]).

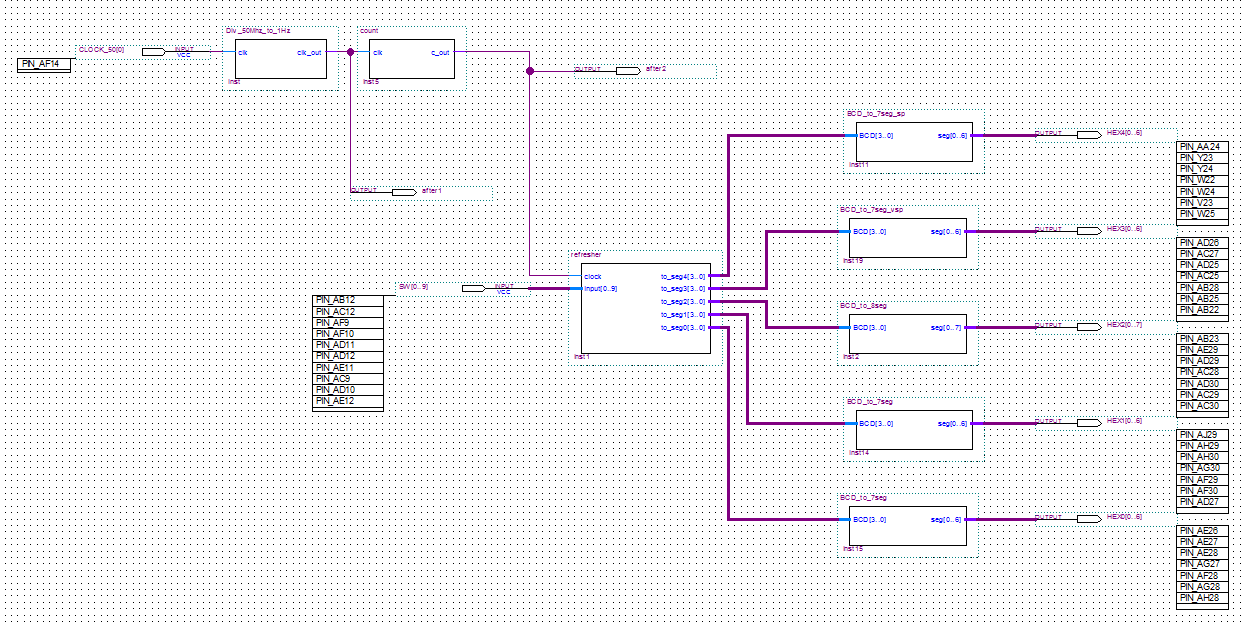


Рисунок 5 - Схема влагомера в Quartus.

# **3 ОТЛАДКА НА ПЛАТЕ**

Для отладки схемы на плате используется *DE1-SoC - Altera Cyclone V 5CSEMA5F31C6N*, под которую велась разработка. Перед отладкой на плате было выполнено моделирование, в котором были рассмотрены всевозможные ситуации на входе и выходе каждого блока. В ходе моделирования логических ошибок выявлено не было. Для отладки на плате требуется при помощи *pin programmer* задать соответствие всем элементам на плате в соответствии с документацией. Для загрузки прошивки на плату использовался *USB*-кабель (*Altera USB Blaster*) и функция *Programmer* среды *Quartus*.

В результате тестирования схема исправно работала согласно заданию (рисунок 6).

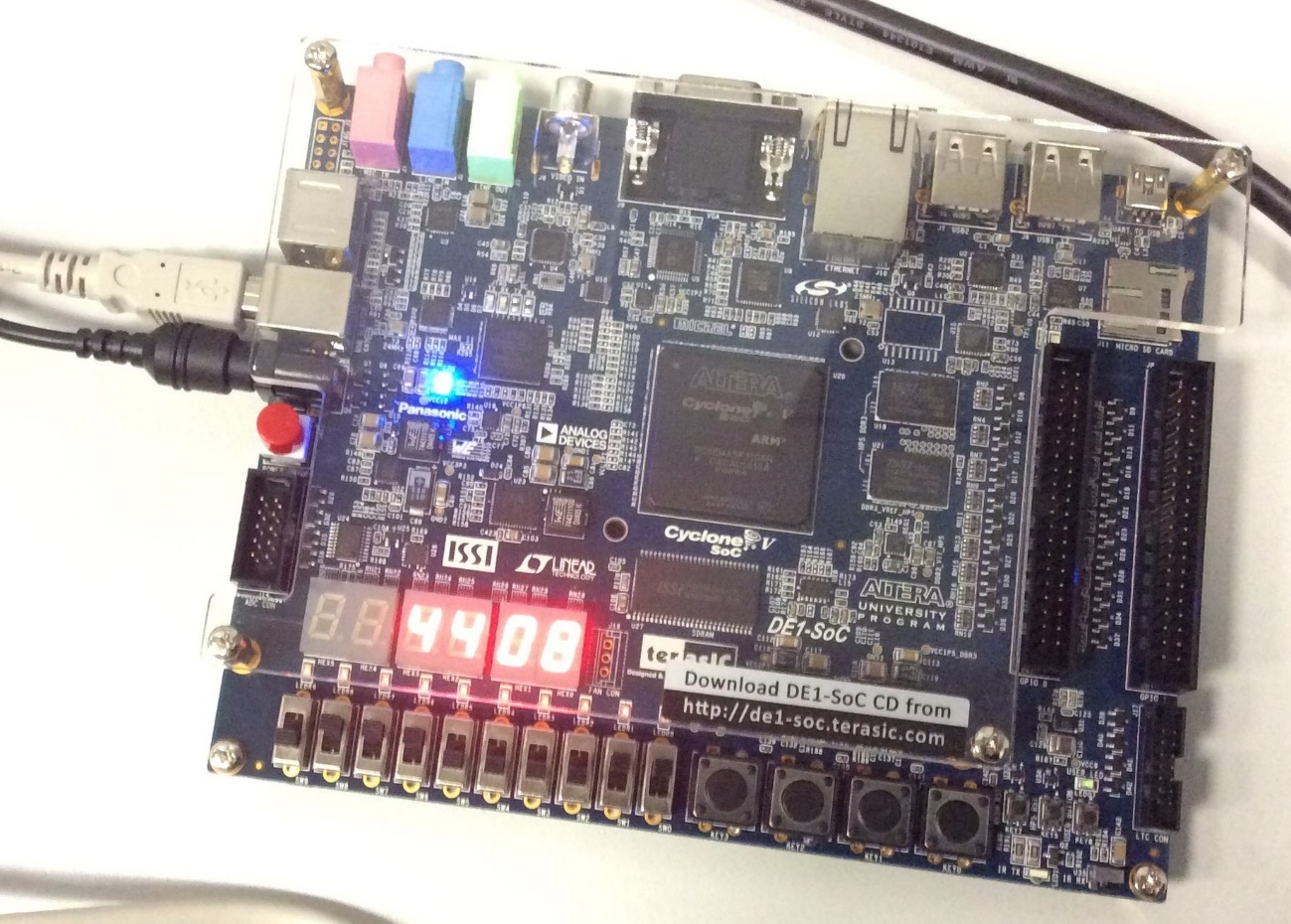


Рисунок 6 – Демонстрация работы схемы.

(число 451 при нормировании в пределах от 0 до 1023 дает 44.08%)

# **ЗАКЛЮЧЕНИЕ**

В ходе выполнения курсовой работы был разработан электронный влагомер с отображением относительной влажности.

Поведение функциональных элементов было описано на языке описания аппаратуры *VHDL*.

Моделирование функционирования узлов и конечной схемы производились в среде *Quartus*, а отладка на макете *DE1-SoC* под управлением *Altera Cyclone V 5CSEMA5F31C6N*.

Разработанная схема электронного влагомера и алгоритм функционирования удовлетворяют требованиям технического задания.

# **СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. Бибило П. Н. Основы языка VHDL. М.: «ЛКИ», 2007. – 328 с.
2. **Суворова Е. А. Шейнин Ю. Е. Проектирование цифровых систем на VHDL. СПб.: «БХВ-Петербург», 2003. – 565 с.**
3. Бибило П. Н. Синтез логических схем с использованием языка VHDL. М.: «СОЛОН-Р», 2002. – 376 с.
4. Ефремов Н. В. Введение в систему автоматизированного проектирования Quartus II: учебное пособие. М.: ГОУ ВПО МГУЛ, 2011. –147 с.
5. Поляков А. К. Языки VHDL и VERILOG в проектировании цифровой аппаратуры на ПЛИС. М.: «ИД МЭИ», 2012. – 220 с.