Министерство образования и науки Российской Федерации

Федеральное агентство по образованию

Государственное образовательное учреждение высшего профессионального образования

«Московский государственный технический университет имени Н.Э. Баумана»

(МГТУ им. Н.Э. Баумана)

Факультет «Информатика и системы управления»

Кафедра «Компьютерные системы и сети»

# Отчет по лабораторной работе №1

по курсу «Организация ЭВМ»

ПРОЕКТИРОВАНИЕ ЦИФРОВЫХ УСТРОЙСТВ НА ОСНОВЕ ПЛИС

*Вариант №5*

Выполнил

студ. гр. ИУ 6-81 \_\_\_\_\_\_\_\_\_\_\_ Головин А.Ю.

Проверил \_\_\_\_\_\_\_\_\_\_\_ Попов А.Ю.

2009

**Цель работы:** закрепление на практике теоретических сведений, полученных при изучении методики проектирования цифровых устройств на основе программируемых логических интегральных схем (ПЛИС), получение необходимых навыков работы с системой автоматизированного проектирования устройств на основе ПЛИС фирмы Xilinx (САПР ISE WebPack 9.1), изучение аппаратных и программных средств моделирования, макетирования и отладки устройств на основе ПЛИС.

**Задание 1.** Выполнить кодирование состояний автомата согласно варианту.

***Таблица 1. Состояния автомата.***

| **Вариант:** | **Двоичный код состояния (Q1Q0)** | | | |
| --- | --- | --- | --- | --- |
| **State0** | **State1** | **State2** | **State3** |
| 9 | 00 | 11 | 10 | 01 |

Определить по таблице 2 функции сигналов управления.

***Таблица 2. Сигналы управления.***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Состояние** | **State0** | **State1** | **State2** | **State3** |
| Двоичный код состояния (Q1Q0) | 00 | 11 | 10 | 01 |
| CNT | 0 | 1 | 1 | 0 |
| CNT\_EN | 0 | 1 | 0 | 1 |
| CNT\_CLR | 1 | 0 | 1 | 0 |

Синтезировать схему автомата. Определить по таблице 3 функции D0=f(Q[20],CNT,Q1,Q0) и D1=f(Q[20],CNT,Q1,Q0).

***Таблица 3. Сигналы D0 и D1.***

| **CNT** | **Q[20]** | **Q1(t)** | **Q0(t)** | **Q1(t+1)** | **Q0(t+1)** | **D1** | **D0** | **Описание события** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | x | 0 | 0 | 0 | 0 | 0 | 0 | Ожидание нажатия кнопки |
| 1 | x | 0 | 0 | 1 | 1 | 1 | 1 | Нажатие кнопки |
| x | 0 | 1 | 1 | 1 | 1 | 1 | 1 | Ожидание окончания счета |
| x | 1 | 1 | 1 | 1 | 0 | 1 | 0 | Конец счета |
| 1 | x | 1 | 0 | 1 | 1 | 1 | 0 | Ожидание отпускания кнопки |
| 0 | x | 1 | 0 | 0 | 1 | 0 | 1 | Отпускание кнопки |
| x | 0 | 0 | 1 | 0 | 0 | 0 | 1 | Ожидание окончания счета |
| x | 1 | 0 | 1 | 0 | 0 | 0 | 0 | Конец счета |

Составим карты Карно.

Для функции D0. Для функции D1.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Q1, Q0**  **CNT, Q20** | **00** | **01** | **11** | **10** |
| **00** | 0 | 0 | 1 | 0 |
| **01** | 0 | 0 | 1 | 0 |
| **11** | 1 | 0 | 1 | 1 |
| **10** | 1 | 0 | 1 | 1 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Q1, Q0**  **CNT, Q20** | **00** | **01** | **11** | **10** |
| **00** | 0 | 1 | 1 | 1 |
| **01** | 0 | 0 | 0 | 1 |
| **11** | 1 | 0 | 0 | 0 |
| **10** | 1 | 1 | 1 | 0 |

Схема автомата представлена на рисунке 1.

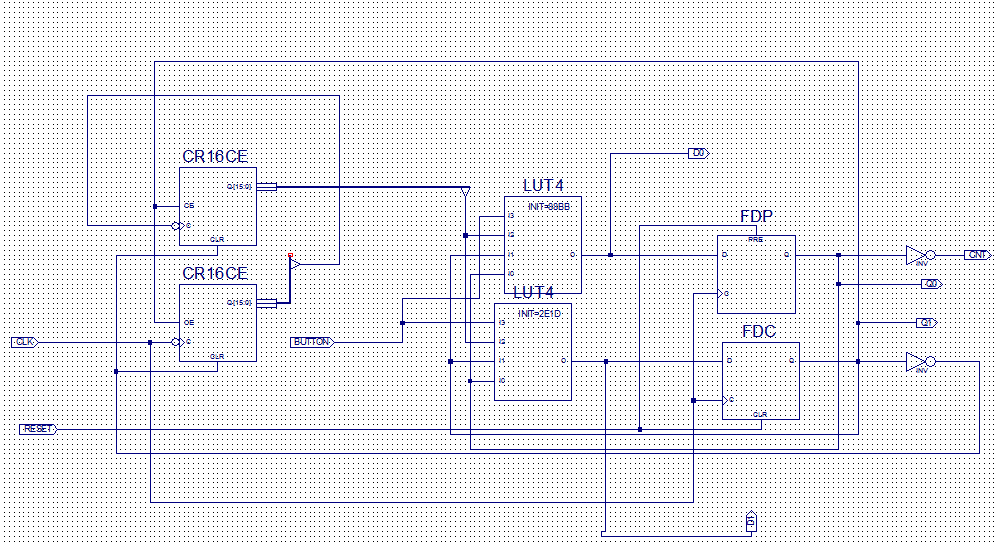


Рисунок 1 – Схема устройства подавления дребезга

Результаты моделирования данного устройства подавления дребезга представлены на рисунке 2.



Рисунок 2 - Моделирование работы схемы подавления дребезга

**Задание 2**. Разработать устройство управления семисегментными индикаторами.

Текст программы на языке VHDL представлен в листинге 1.

***Листинг 1. Устройство управления индикаторами.***

**LIBRARY** ieee;

**USE** ieee.std\_logic\_1164.**ALL**;

**USE** ieee.std\_logic\_arith.**ALL**;

**ENTITY** Seven\_Segment\_Driver **IS**

**PORT**(

clk\_div : **IN** std\_logic;

Q : **IN** std\_logic\_vector(15 **DOWNTO** 0);

rst : **IN** std\_logic;

D : **OUT** std\_logic\_vector(3 **DOWNTO** 0);

A : **INOUT** std\_logic\_vector(3 **DOWNTO** 0) );

**END** **ENTITY** Seven\_Segment\_Driver;

**ARCHITECTURE** Struct **OF** Seven\_Segment\_Driver **IS**

**BEGIN**

A\_drive: **PROCESS** (clk\_div,rst)

**BEGIN**

I**F** (rst = '1') **THEN**

A<="1110";

**ELSIF** (clk\_div'**EVENT** **AND** clk\_div='1') **THEN**

A(3)<=A(2);

A(2)<=A(1);

A(1)<=A(0);

A(0)<=A(3);

**END** **IF**;

**END** **PROCESS** A\_drive;

D(0) <= (Q(0) **AND** **NOT**(A(0)))

OR (Q(4) **AND** **NOT**(A(1)))

OR (Q(8) **AND NOT**(A(2)))

OR (Q(12) **AND NOT**(A(3)));

D(1) <= (Q(1) **AND NOT**(A(0)))

OR (Q(5) **AND NOT**(A(1)))

OR (Q(9) **AND NOT**(A(2)))

OR (Q(13) **AND NOT**(A(3)));

D(2) <= (Q(2) **AND NOT**(A(0)))

OR (Q(6) **AND NOT**(A(1)))

OR (Q(10) **AND NOT**(A(2)))

OR (Q(14) **AND NOT**(A(3)));

D(3) <= (Q(3) **AND NOT**(A(0)))

OR (Q(7) **AND NOT**(A(1)))

OR (Q(11) **AND NOT**(A(2)))

OR (Q(15) **AND NOT**(A(3)));

**END** **ARCHITECTURE** Struct;

Результаты моделирования представлены на рисунке 3.

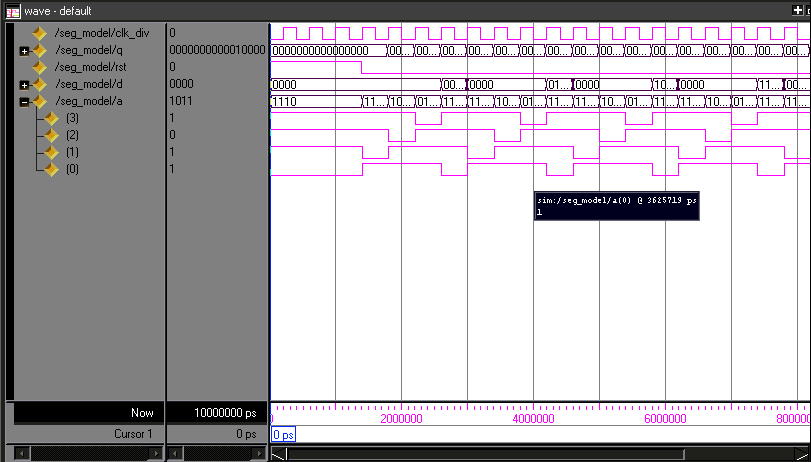


Рисунок 3 – Моделирование устройства управления

**Задание 3.** Разработать поведенческое описание схемы преобразования кода.

Текст схемы преобразователя четырехразрядного кода в код управления индикаторами представлен в листинге 2.

***Листинг 2. Описание преобразователя кода.***

**library** IEEE;

**use** IEEE.STD\_LOGIC\_1164.ALL;

**use** IEEE.STD\_LOGIC\_ARITH.ALL;

**use** IEEE.STD\_LOGIC\_UNSIGNED.ALL;

**ENTITY** Decode\_Ind **IS**

**PORT** ( DH : **IN** std\_logic\_vector (3 **downto** 0);

SEG\_DATA : **OUT** std\_logic\_vector (7 **downto** 0)

);

**END** Decode\_Ind;

**ARCHITECTURE** Behavioral **OF** Decode\_Ind **IS**

**BEGIN**

**PROCESS** (DH)

**BEGIN**

**CASE** DH **IS**

**WHEN** "0000" => SEG\_DATA <= "10000001";

**WHEN** "0001" => SEG\_DATA <= "11001111";

**WHEN** "0010" => SEG\_DATA <= "10010010";

**WHEN** "0011" => SEG\_DATA <= "10000110";

**WHEN** "0100" => SEG\_DATA <= "11001100";

**WHEN** "0101" => SEG\_DATA <= "10100100";

**WHEN** "0110" => SEG\_DATA <= "10100000";

**WHEN** "0111" => SEG\_DATA <= "10001111";

**WHEN** "1000" => SEG\_DATA <= "10000000";

**WHEN** "1001" => SEG\_DATA <= "10000100";

**WHEN** "1010" => SEG\_DATA <= "10001000";

**WHEN** "1011" => SEG\_DATA <= "11100000";

**WHEN** "1100" => SEG\_DATA <= "10110001";

**WHEN** "1101" => SEG\_DATA <= "11000010";

**WHEN** "1110" => SEG\_DATA <= "10110000";

**WHEN** "1111" => SEG\_DATA <= "10111000";

**WHEN** **OTHERS** => null;

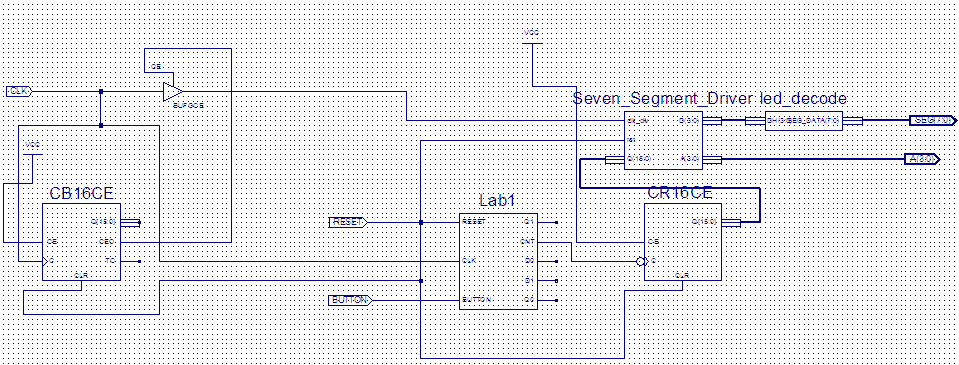
**END** **CASE**;

**END** **PROCESS**;

**END** Behavioral;

**Задание 4.** Собрать модель полученного устройства.

Модель полученного устройства представлена на рисунке 4.



***Рисунок 4 - Модель полученного устройства.***