

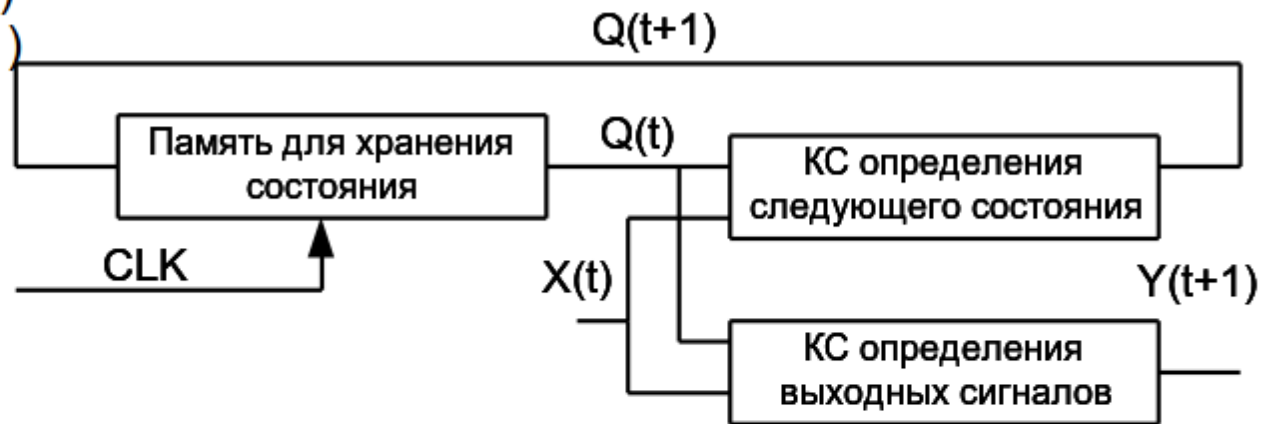
Семинар 5

Проектирование устройств
управления с жесткой
логикой

Автомат Мили

Схема автомата Мили

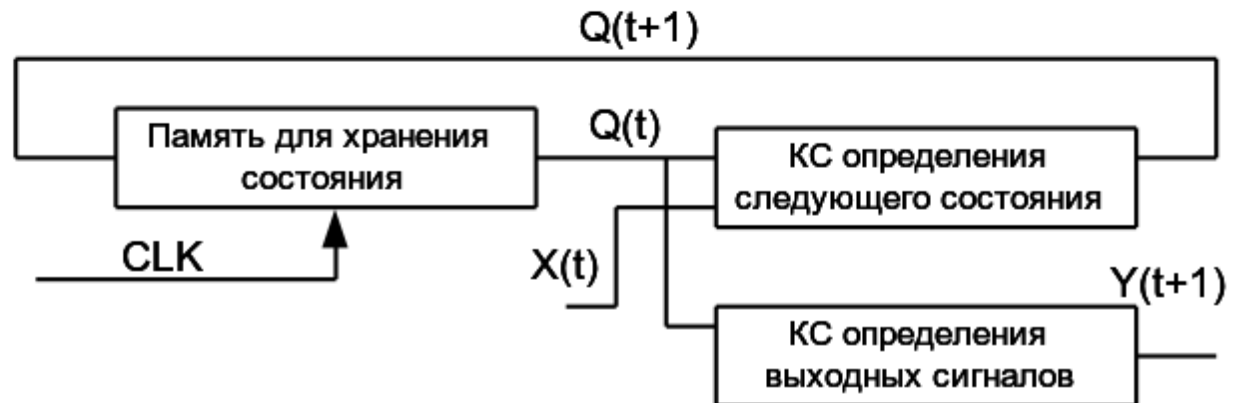
$$\begin{cases} Q(t+1) = A (Q(t), x(t)) \\ Y(t+1) = B (Q(t), x(t)) \end{cases}$$



Автомат Мура

Схема автомата Мура

$$\begin{cases} Q(t+1) = A (Q(t), x(t)) \\ Y(t+1) = B (Q(t)) \end{cases}$$



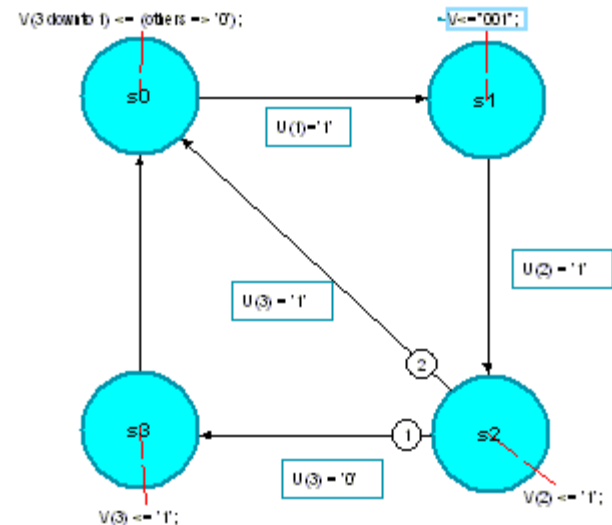
Описание автомата Мура на языке VHDL

(вариант с синхронными входами и выходами)

```

LIBRARY ieee;
USE ieee.std_logic_1164.all;
USE ieee.std_logic_arith.all;
ENTITY control_unit IS
    PORT(U : IN    std_logic_vector ( 3 DOWNT0 1 );
          clk : IN    std_logic;
          rst : IN    std_logic;
          V : OUT   std_logic_vector ( 3 DOWNT0 1 ) );
END control_unit;
ARCHITECTURE moore OF control_unit IS
    TYPE STATE_TYPE IS (s0, s1,s2,s3);
    SIGNAL current_state : STATE_TYPE;
BEGIN
    clocked_proc : PROCESS (clk, rst)
    BEGIN
        IF (rst = '0') THEN
            current_state <= s0;
        ELSIF (clk'EVENT AND clk = '1') THEN

```

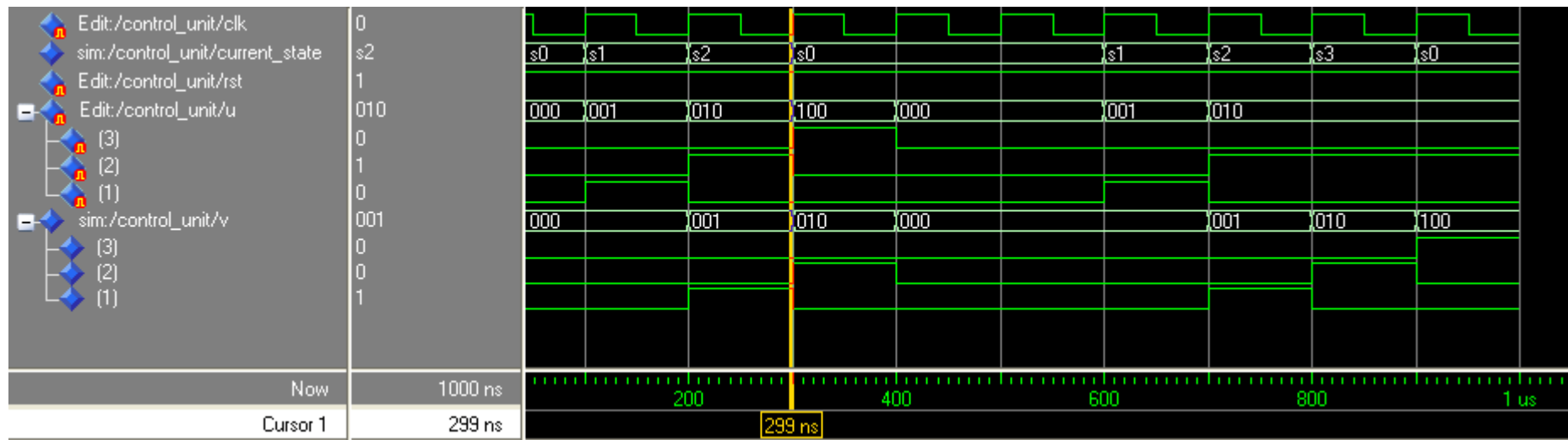


```

CASE current_state IS
    WHEN s0 =>
        V(3 downto 1) <= (others => '0');
        IF (U(1)='1') THEN current_state <= s1;
        ELSE current_state <= s0; END IF;
    WHEN s1 =>
        V<= "001";
        IF (U(2) = '1') THEN current_state <= s2;
        ELSE current_state <= s1; END IF;
    WHEN s2 =>
        V <= "010";
        IF (U(3) = '0') THEN current_state <= s3;
        ELSE current_state <= s0; END IF;
    WHEN s3 =>
        V <= "100";
        current_state <= s0;
    WHEN OTHERS =>
        current_state <= s0;
END CASE;
END IF;
END PROCESS clocked_proc;
END moore;

```

Тест автомат Мура в ModelSim 6



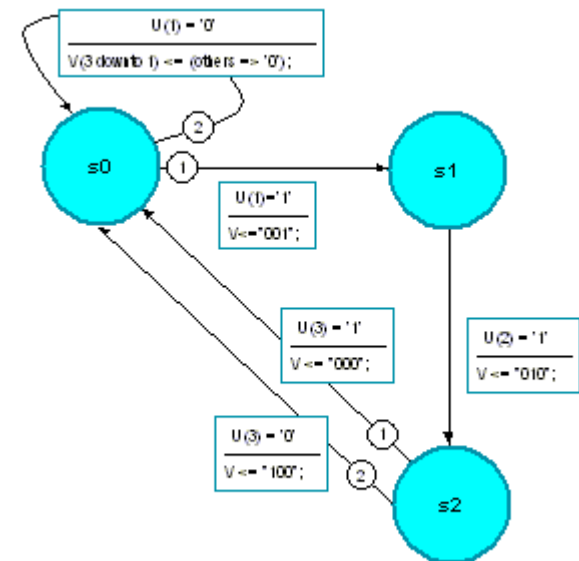
Описание автомата Мили на языке VHDL

(вариант с синхронными выходами)

```

LIBRARY ieee;
USE ieee.std_logic_1164.all;
USE ieee.std_logic_arith.all;
ENTITY control_unit IS
    PORT(U : IN    std_logic_vector ( 3 DOWNT0 1 );
          clk : IN    std_logic;
          rst : IN    std_logic;
          V : OUT    std_logic_vector ( 3 DOWNT0 1 ) );
END control_unit;
ARCHITECTURE mielie OF control_unit IS
    TYPE STATE_TYPE IS (s0, s1,s2);
    SIGNAL current_state : STATE_TYPE;
BEGIN
    clocked_proc : PROCESS (clk, rst)
    BEGIN
        IF (rst = '0') THEN
            current_state <= s0;
        ELSIF (clk'EVENT AND clk = '1') THEN

```



Описание автомата Мили на языке VHDL (окончание)

```
CASE current_state IS
```

```
  WHEN s0 =>
```

```
    IF (U(1)='1') THEN
```

```
      V<="001";
```

```
      current_state <= s1;
```

```
    ELSIF (U(1) = '0') THEN
```

```
      V(3 downto 1) <= (others =>
```

```
'0');
```

```
      current_state <= s0;
```

```
    ELSE
```

```
      current_state <= s0;
```

```
    END IF;
```

```
  WHEN s1 =>
```

```
    IF (U(2) = '1') THEN
```

```
      V <= "010";
```

```
      current_state <= s2;
```

```
    ELSE
```

```
      current_state <= s1;
```

```
    END IF;
```

```
  WHEN s2 =>
```

```
    IF (U(3) = '1') THEN
```

```
      V <= "000";
```

```
      current_state <= s0;
```

```
    ELSIF (U(3) = '0') THEN
```

```
      V <= "100";
```

```
      current_state <= s0;
```

```
    ELSE
```

```
      current_state <= s2;
```

```
    END IF;
```

```
  WHEN OTHERS =>
```

```
    current_state <= s0;
```

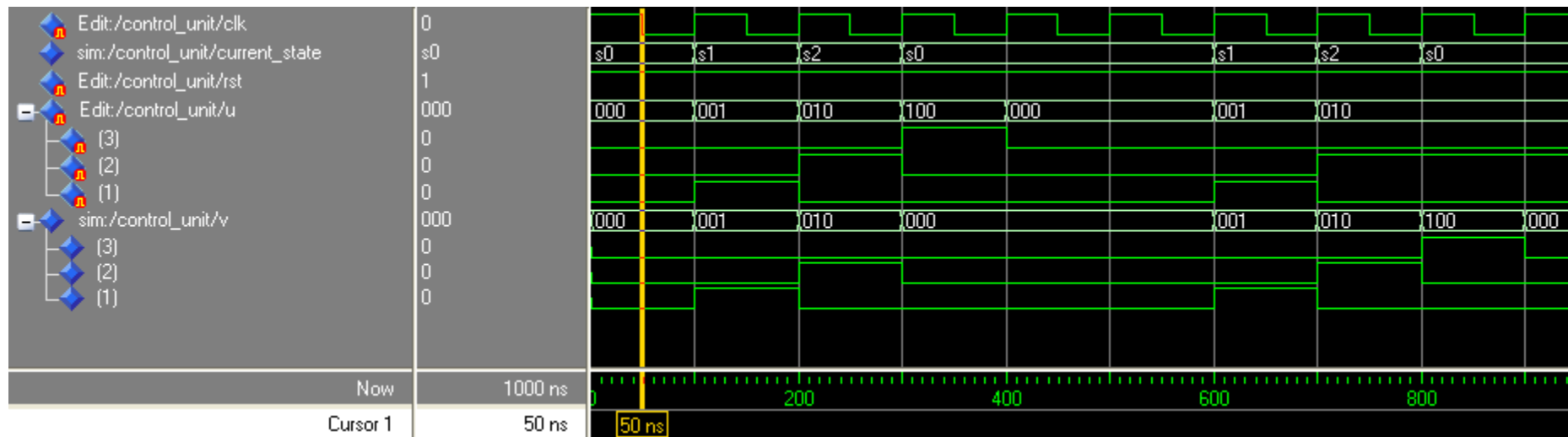
```
  END CASE;
```

```
  END IF;
```

```
END PROCESS clocked_proc;
```

```
END mielie;
```

Тест автомат Мили в ModelSim 6



Домашнее задание по курсу «ЭВМ»

В ходе выполнения домашнего задания необходимо разработать устройство управления схемного типа, обрабатывающий входное командное слово $C=\{ABCDEF\}$ и выдающий сигналы управления $M=\{M_0, \dots, M_{k-1}\}$ операционному блоку в соответствии с приведенной в индивидуальном задании логикой работы.

Этап 1.

А. По диаграмме переходов автомата (Приложение 1) и описанию условий переходов и активных сигналов (дополнительный файл варианты.pdf), определить тип управляющего автомата (автомат Мили или Мура, смешанный). Выбор обосновать.

В. Произвести кодирование состояний управляющего автомата. Составить схему переходов/состояний полученного автомата. Схему представить в отчете.

Этап 2.

Разработать описание устройства управления на языке VHDL, для чего использовать приведенные в Приложении 2 шаблоны для автоматов Мили и Мура.

Разработать тестовое описание для устройства, представляющее собой генератор входных сигналов (см. Приложение 3). Тестовое описание должно обеспечивать проверку всех ветвей автомата.

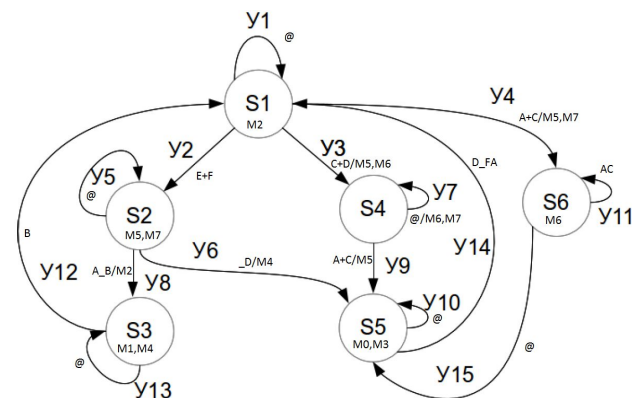
Этап 3.

А. Установить ПО ModelSim PE (или аналогичный продукт: Xilinx ISE, Altera Quartus).

Б. Выполнить моделирование полученного теста в ПО ModelSim PE. Результаты моделирования представить в отчете.

Описание устройства управления

Тестовое описание



Варианты индивидуальных заданий по курсу ЭВМ

Таблица 1. Варианты диаграмм и активных сигналов.

Вариант	Диаграмма переходов	Активные сигналы М в состоянии					
		S1	S2	S3	S4	S5	S6
1.	1	-	-	-	-	-	-
2.	2	-	1	-	4, 5, 6	0,2	3,7
3.	3	-	0,2	7	1,3	4,5	6

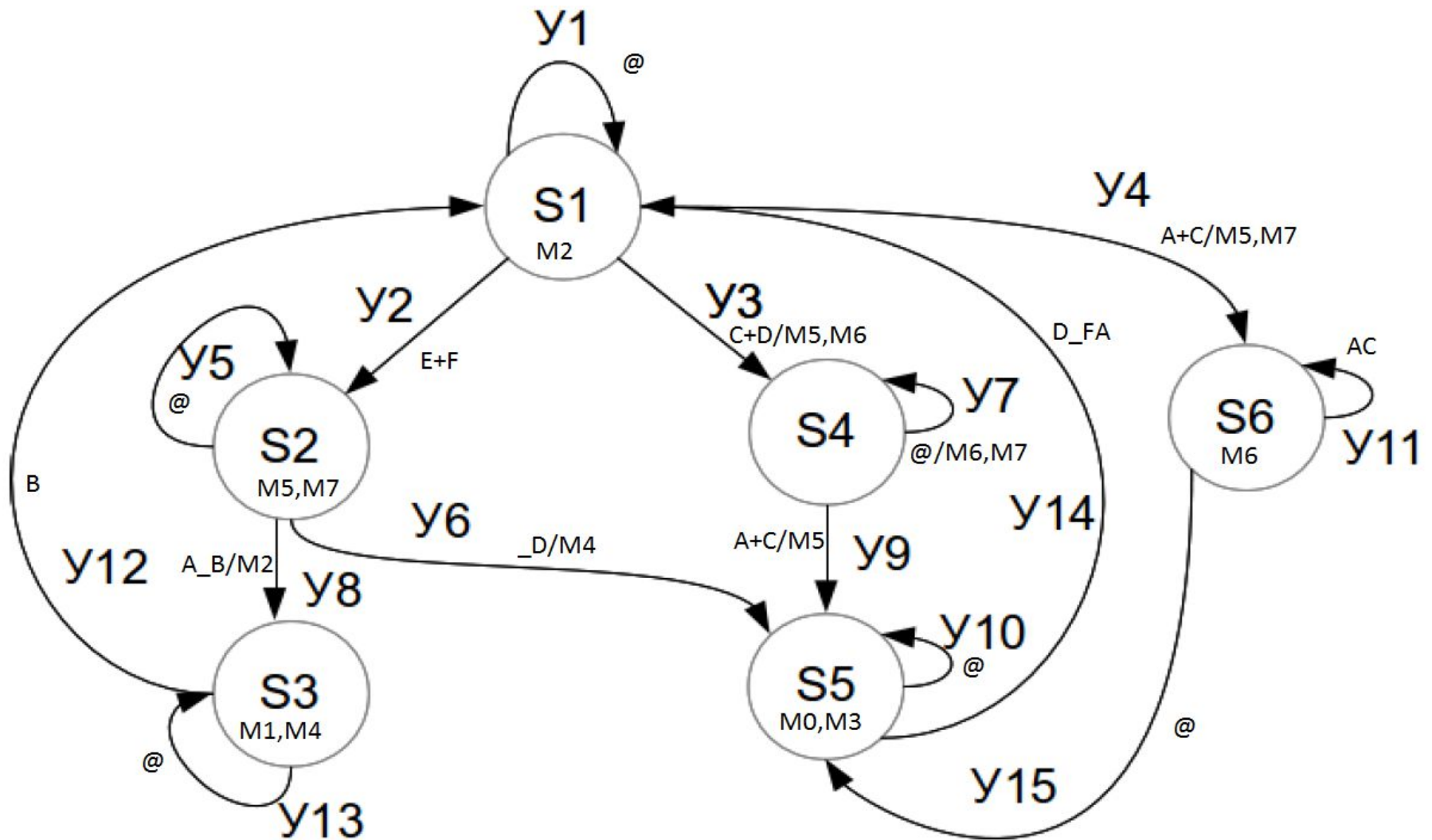
Таблица 2. Условия переходов и наименование отладочной платы («@» - иначе, «_X» - НЕ X, «+» - ИЛИ, 1- безусловный переход).

Вариант	Название отладочной платы	Активные сигналы М в состоянии														
		Y1	Y2	Y3	Y4	Y5	Y6	Y7	Y8	Y9	Y10	Y11	Y12	Y13	Y14	Y15
1.	Spartan3	@	ABC	A_B	@	E+D	F	@	EF	@	A+_C	_A	AB	@	@	1
2.	Nexus2	@	CD_E	A+_C	@	E+C	F	@	D_F	B	@	_AC	AB	@	@	ABCF

Таблица 3. Активные сигналы для переходов.

Вариант	Активные сигналы М в состоянии														
	Y1	Y2	Y3	Y4	Y5	Y6	Y7	Y8	Y9	Y10	Y11	Y12	Y13	Y14	Y15
1.	0	-	-	1,2	0,3	-	3	3	6,7	-	2,4	-	2	3,5	-

Диаграмма переходов состояний управляющего автомата



Листинг VHDL описания управляющего автомата

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;

entity control_unit is
    Port ( C : in STD_LOGIC_VECTOR (5 downto 0);
          clk : in STD_LOGIC;
          rst : in STD_LOGIC;
          M : out STD_LOGIC_VECTOR (7 downto 0));
end control_unit;

architecture hybrid of control_unit is
    type STATE_TYPE is (s1,s2,s3,s4,s5,s6);
    signal cur_state : STATE_TYPE;
begin
    clocked_proc : process (clk,rst)
    begin
        if (rst = '0') then
            cur_state <= s1;
            M <= "00000000";
        elsif (clk'event and clk = '1') then
            case cur_state is
                when s1 =>
                    M <= "00000100"; -- M2
                    if (C(4) = '1' or C(5) = '1') then -- E or F
                        cur_state <= s2;
                    elsif (C(2) = '1' or C(3) = '1') then -- C or D
                        M(5) <= '1';
                        M(6) <= '1';
                        cur_state <= s4;
                    elsif (C(0) = '1' or C(2) = '1') then -- A or C
                        M(5) <= '1';
                        M(7) <= '1';
                        cur_state <= s6;
                    else -- @
                        cur_state <= s1;
                    end if;
                when s2 =>
                    M <= "10100000"; -- M5, M7
                    if (C(3) = '0') then -- not D
                        M(4) <= '1';
                        cur_state <= s5;
                    end if;
                when s3 =>
                    M <= "00000000";
                    cur_state <= s3;
                when s4 =>
                    M <= "00000000";
                    cur_state <= s4;
                when s5 =>
                    M <= "00000000";
                    cur_state <= s5;
                when s6 =>
                    M <= "00000000";
                    cur_state <= s6;
            end case;
        end if;
    end process;
end hybrid;
```

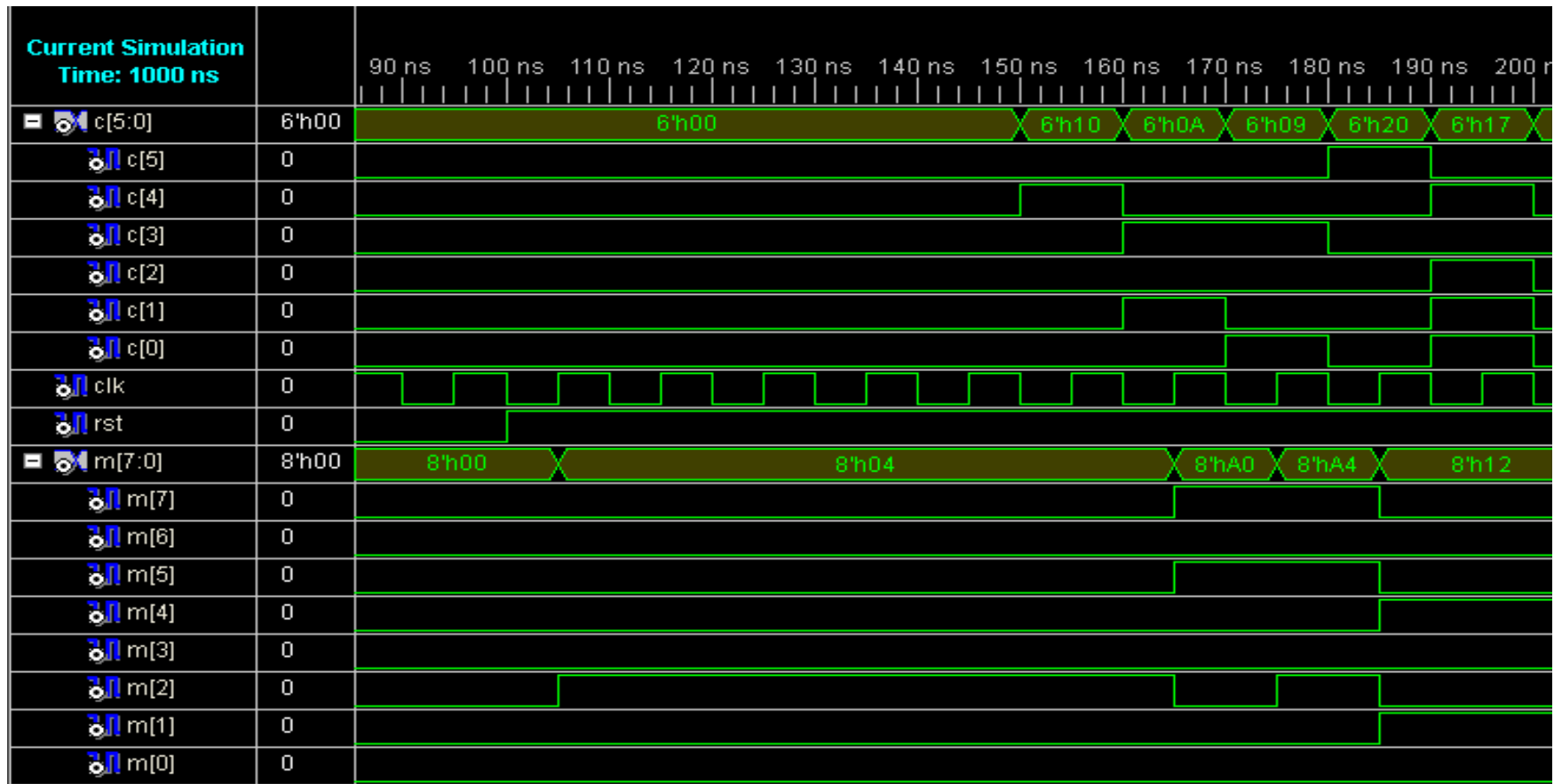
```

        elsif (C(0) = '1' and C(1) = '0') then -- A & _B
            M(2) <= '1';
            cur_state <= s3;

        else -- @
            cur_state <= s2;
        end if;
    when s3 =>
        M <= "00010010"; -- M1, M4
        if (C(1) = '1') then -- B
            cur_state <= s1;
        else -- @
            cur_state <= s3;
        end if;
    when s4 =>
        M <= "00000000";
        if (C(0) = '1' or C(2) = '1') then -- A or C
            M(5) <= '1';
            cur_state <= s5;
        else -- @
            M(6) <= '1';
            M(7) <= '1';
            cur_state <= s4;
        end if;
    when s5 =>
        M <= "00001001"; -- M0, M3
        if (C(3) = '1' and C(5) = '0'
            and C(0) = '1') then -- D & _F & A
            cur_state <= s1;
        else -- @
            cur_state <= s5;
        end if;
    when s6 =>
        M <= "01000000"; -- M6
        if (C(0) = '1' and C(2) = '1') then -- A and C
            cur_state <= s6;
        else -- @
            cur_state <= s5;
        end if;
    when others =>
        cur_state <= s1;
    end case;
end if;

```

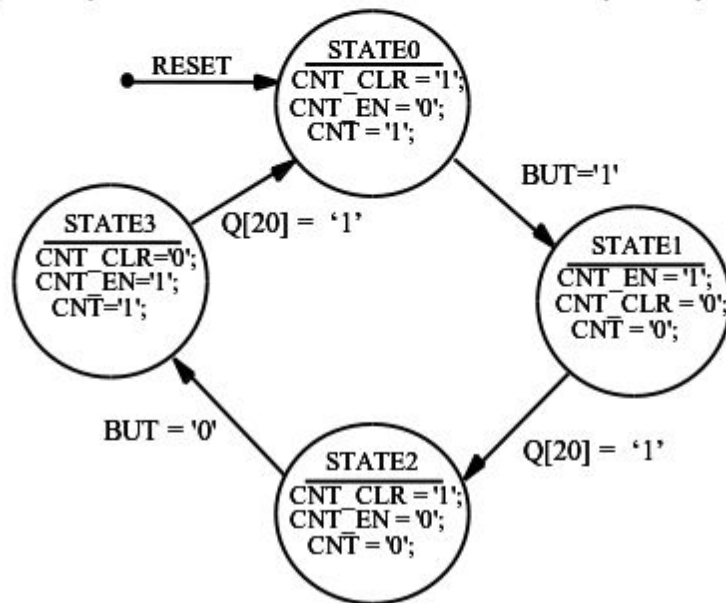
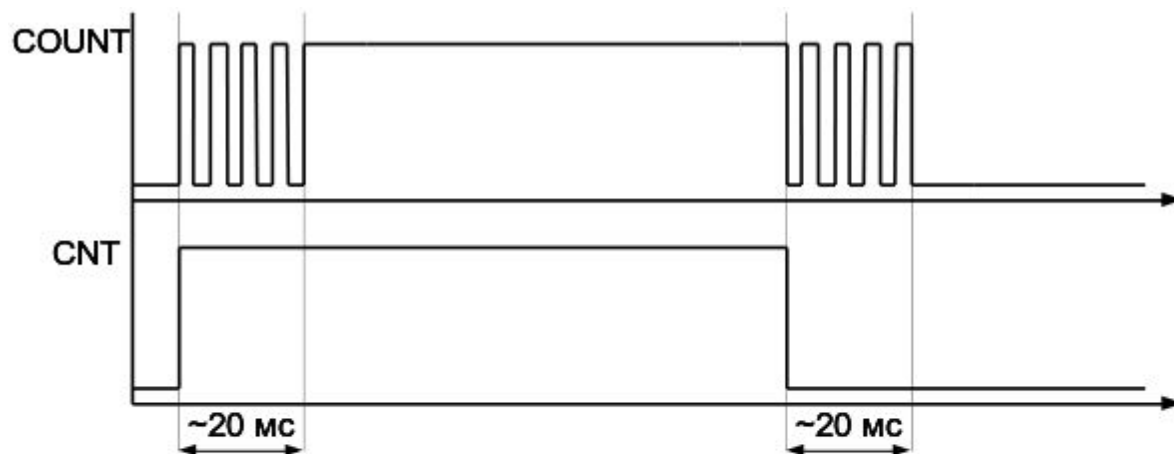
Моделирование



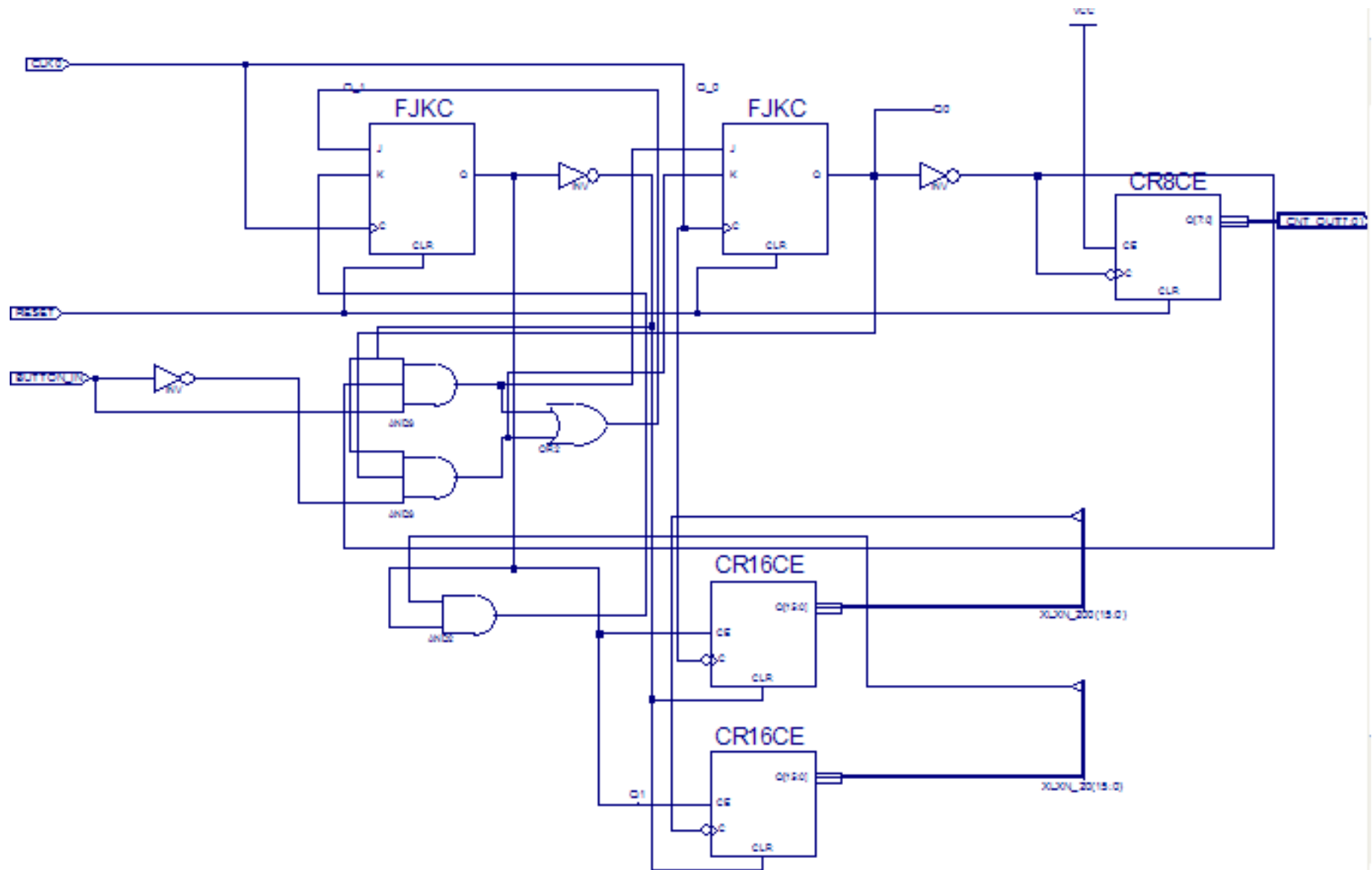
Пример 1

Устройство подавления
дребезга
(Структурный вариант)

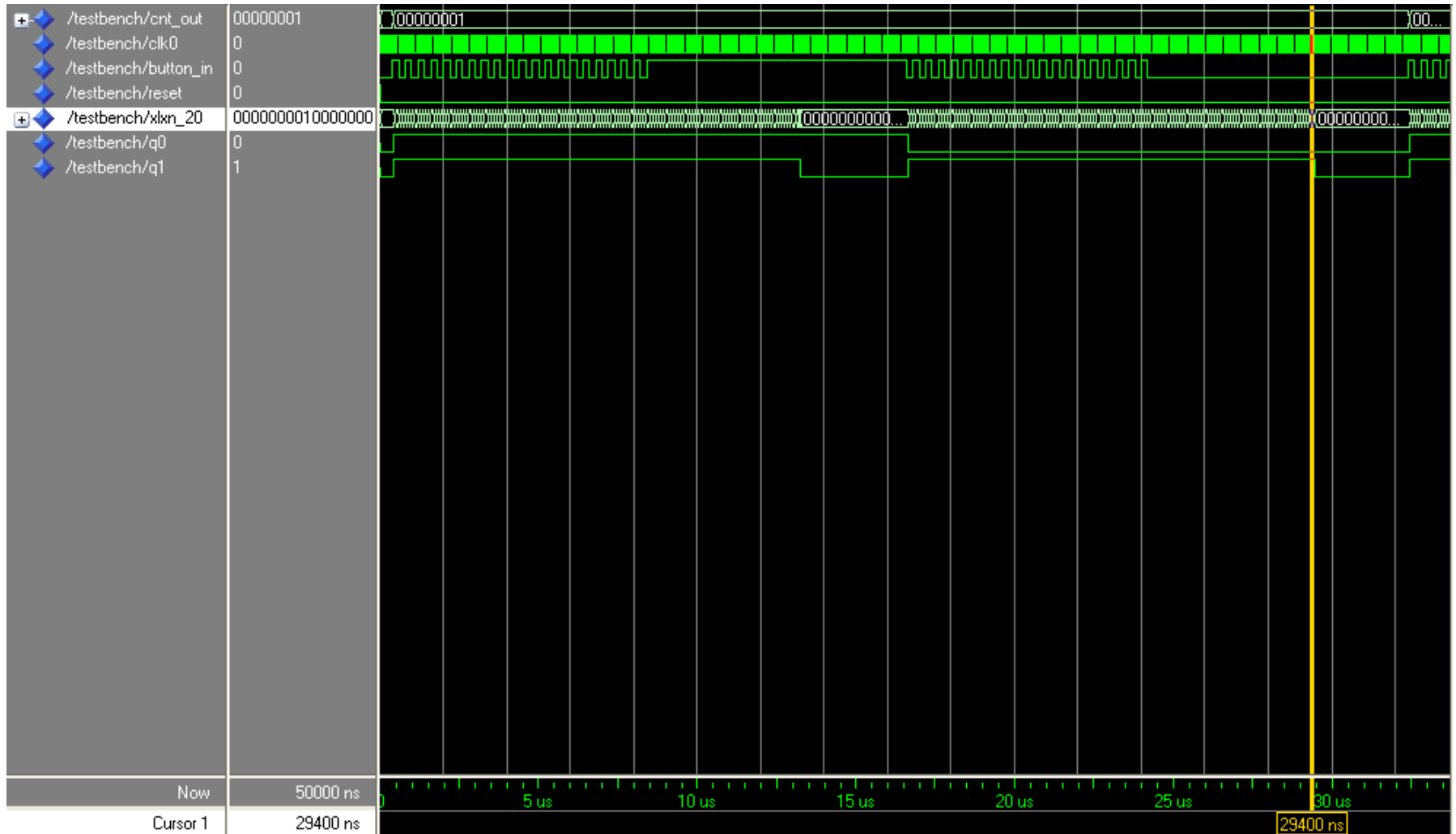
Диаграмма работы устройства



Этап ввода схемы устройства



□ Тестирование модели в ModelSim 6.1



Настройка блоков ввода/вывода в редакторе ресурсов (PACE)

Xilinx PACE - C:\LABS\my_project\drebezg\dREBEZG_STATECAD\dREBEZG_STATECADV1.ucf

File Edit View IOBs Areas Tools Window Help

Design Browser

- I/O Pins
- Global Logic
- Logic

Design Object List - I/O Pins

	I/O Name	I/O Direction	Loc	Bank	I/O Std.	Vref	Vcco	Drive Str.	T
	BUTTON_IN	Input	M13	BANK3					
	CLK0	Input	T9	BANK4					
	CNT_OUT<0>	Output	K12	BANK3					
	CNT_OUT<1>	Output	P14	BANK3					
	CNT_OUT<2>	Output	L12	BANK3					
	CNT_OUT<3>	Output	N14	BANK3					
	CNT_OUT<4>	Output	P13	BANK4					
	CNT_OUT<5>	Output	N12	BANK4					
	CNT_OUT<6>	Output	P12	BANK4					
	CNT_OUT<7>	Output	P11	BANK4					

#	Group	I/O Direction	Loc	I/O Std.	Vref	Vcco	Drive Str.	Termination	Slew	Delay	Swap	Local C
8	CNT_O	Output									No	

Device Architecture for xc3s200-4-ft256

Package View Architecture View

X3Y14

Общий отчет по проектированию схемы устранения дребезга (структурный вариант)

The screenshot displays the Xilinx ISE Design Summary window for a project named 'But_driver_sch'. The window is divided into several panes. The 'Sources' pane on the left shows the project hierarchy: 'But_driver_sch' containing 'xc3s200-4ft256' and 'Schematic1 (Schematic1.sch)'. The 'Processes' pane shows the design flow steps, with 'View Design Summary' selected. The main area displays the 'Device Utilization Summary' and 'Performance Summary' tables.

Device Utilization Summary

Logic Utilization	Used	Available	Utilization	Note(s)
Number of Slice Flip Flops	32	3,840	1%	
Number of 4 input LUTs	2	3,840	1%	
Logic Distribution				
Number of occupied Slices	32	1,920	1%	
Number of Slices containing only related logic	32	32	100%	
Number of Slices containing unrelated logic	0	32	0%	
Total Number of 4 input LUTs	2	3,840	1%	
Number of bonded IOBs	11	173	6%	
Number of GCLKs	1	8	12%	
Number of RPM macros	2			
Total equivalent gate count for design	271			
Additional JTAG gate count for IOBs	528			

Performance Summary

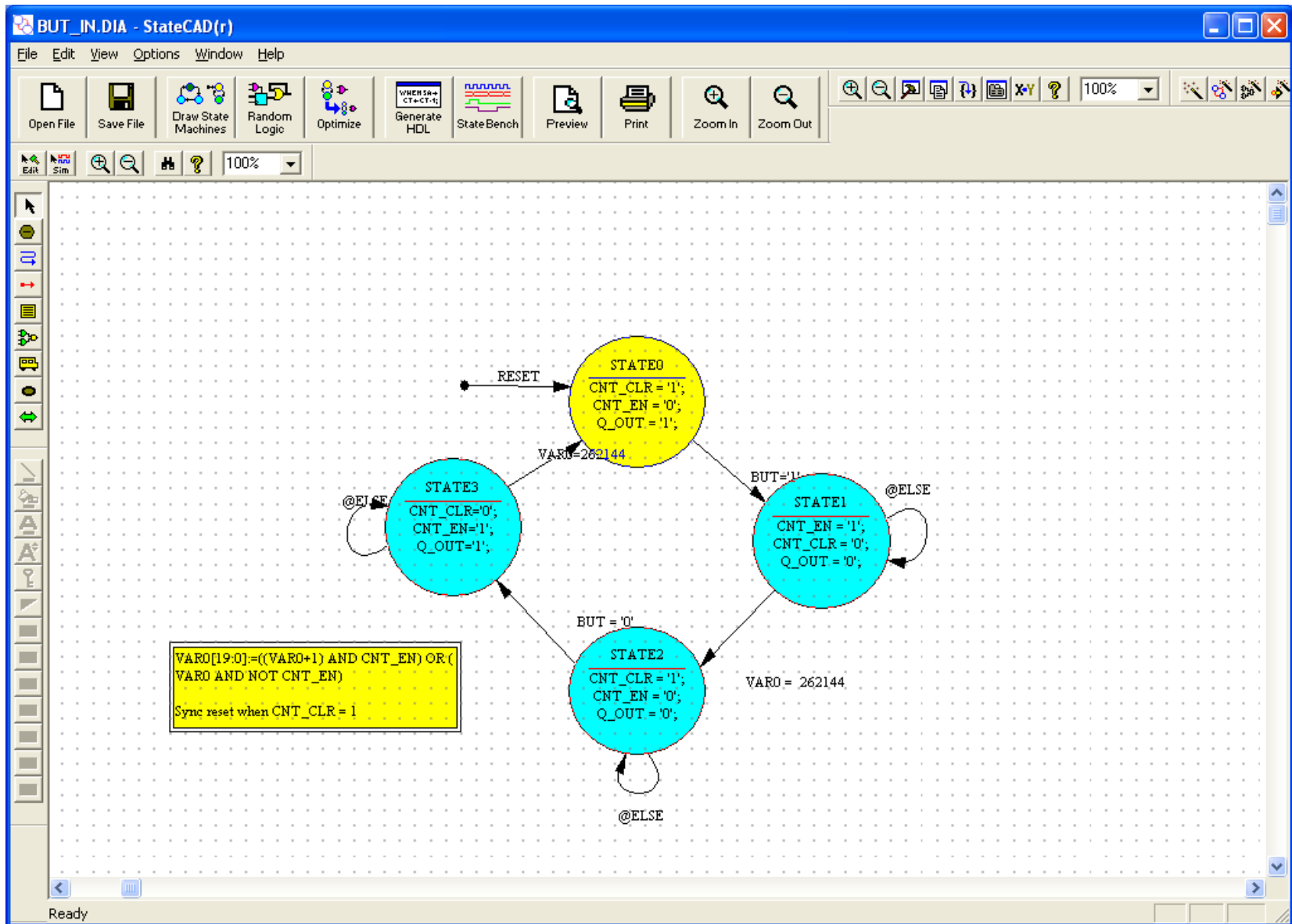
Final Timing Score:	0	Pinout Data:	Pinout Report
Routing Results:	All Signals Completely Routed	Clock Data:	Clock Report
Timing Constraints:	All Constraints Met		

The 'Transcript' pane at the bottom shows the message: 'Started : "Launching Design Summary".'

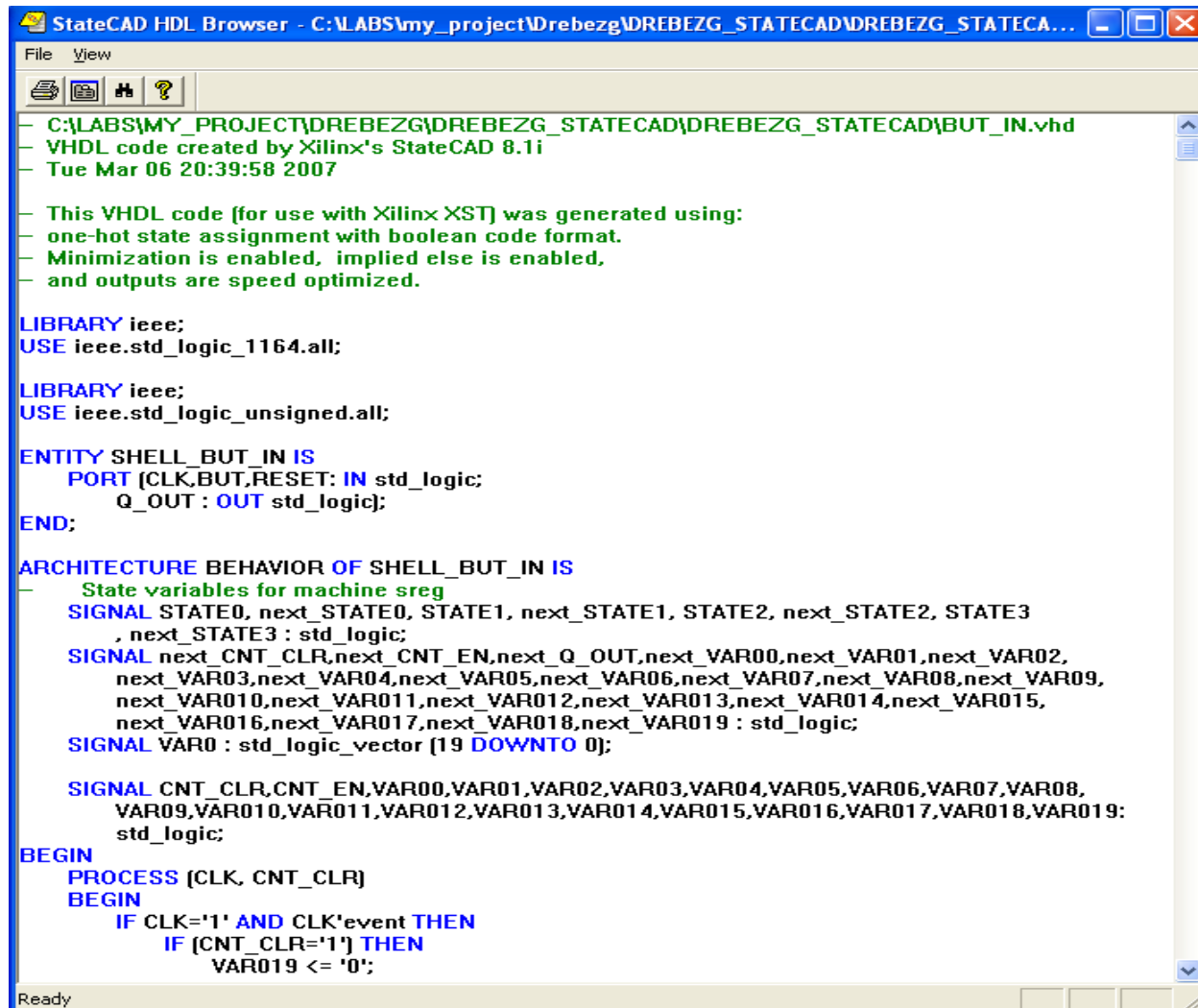
▫ Пример 2

- Устройство подавления дребезга
 - (Поведенческий вариант)

Этап ввода диаграммы автомата



Генерация VHDL описания автомата



The screenshot shows the StateCAD HDL Browser window. The title bar reads "StateCAD HDL Browser - C:\LABS\my_project\Drebezg\DREBEZG_STATECAD\DREBEZG_STATECA...". The menu bar has "File" and "View". The toolbar contains icons for file operations and help. The main text area displays the following VHDL code:

```
-- C:\LABS\MY_PROJECT\DREBEZG\DREBEZG_STATECAD\DREBEZG_STATECAD\BUT_IN.vhd
-- VHDL code created by Xilinx's StateCAD 8.1i
-- Tue Mar 06 20:39:58 2007

-- This VHDL code (for use with Xilinx XST) was generated using:
-- one-hot state assignment with boolean code format.
-- Minimization is enabled, implied else is enabled,
-- and outputs are speed optimized.

LIBRARY ieee;
USE ieee.std_logic_1164.all;

LIBRARY ieee;
USE ieee.std_logic_unsigned.all;

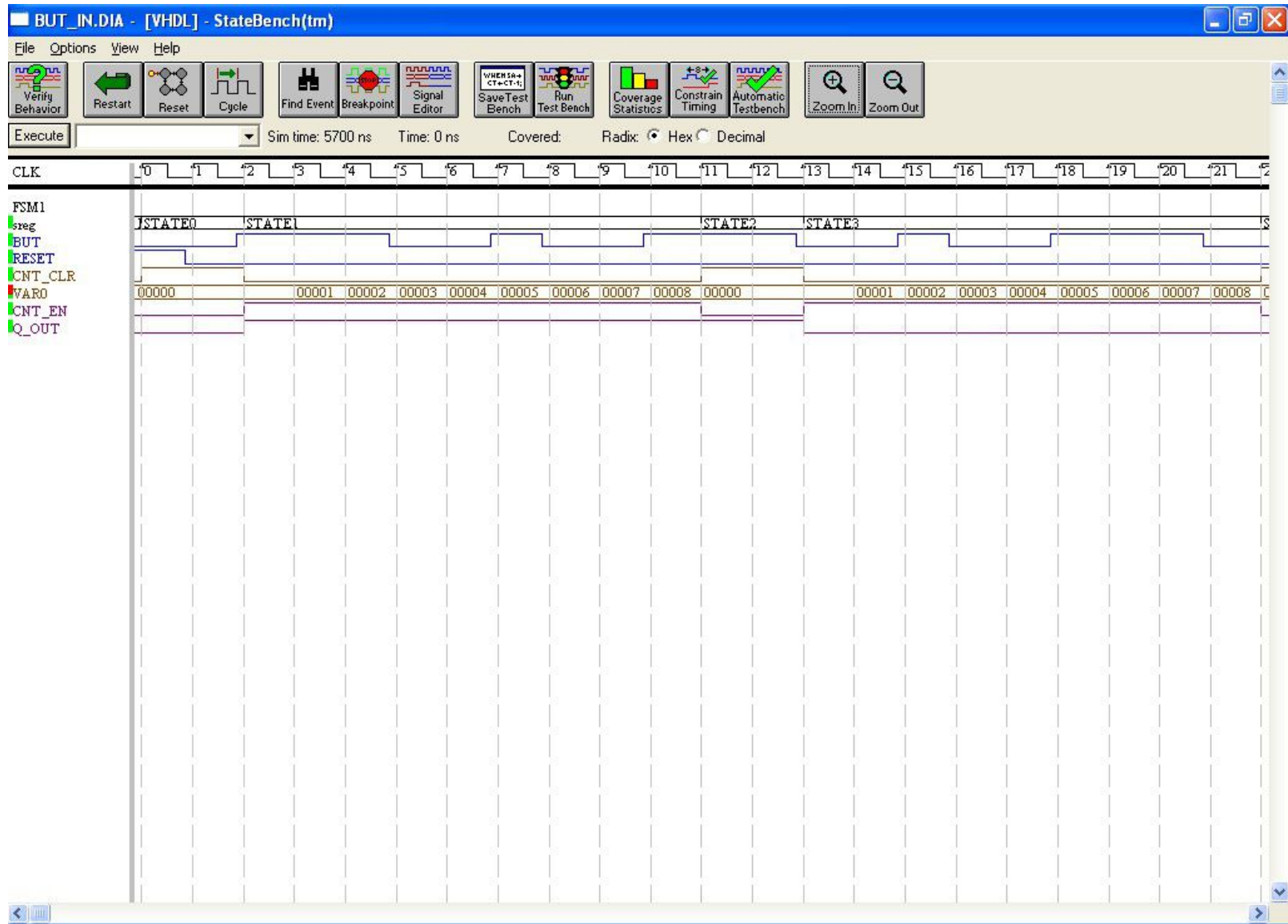
ENTITY SHELL_BUT_IN IS
    PORT (CLK,BUT,RESET: IN std_logic;
          Q_OUT : OUT std_logic);
END;

ARCHITECTURE BEHAVIOR OF SHELL_BUT_IN IS
-- State variables for machine sreg
    SIGNAL STATE0, next_STATE0, STATE1, next_STATE1, STATE2, next_STATE2, STATE3
        , next_STATE3 : std_logic;
    SIGNAL next_CNT_CLR,next_CNT_EN,next_Q_OUT,next_VAR00,next_VAR01,next_VAR02,
        next_VAR03,next_VAR04,next_VAR05,next_VAR06,next_VAR07,next_VAR08,next_VAR09,
        next_VAR010,next_VAR011,next_VAR012,next_VAR013,next_VAR014,next_VAR015,
        next_VAR016,next_VAR017,next_VAR018,next_VAR019 : std_logic;
    SIGNAL VAR0 : std_logic_vector (19 DOWNTO 0);

    SIGNAL CNT_CLR,CNT_EN,VAR00,VAR01,VAR02,VAR03,VAR04,VAR05,VAR06,VAR07,VAR08,
        VAR09,VAR010,VAR011,VAR012,VAR013,VAR014,VAR015,VAR016,VAR017,VAR018,VAR019:
        std_logic;
BEGIN
    PROCESS (CLK, CNT_CLR)
    BEGIN
        IF CLK='1' AND CLK'event THEN
            IF (CNT_CLR='1') THEN
                VAR019 <= '0';
```

The status bar at the bottom left shows "Ready".

Тестирование модели автомата в StateCad



Описание основного модуля

The screenshot displays the Xilinx ISE environment for a project named "DREBEZG_STATECAD". The main workspace shows a logic diagram with the following components and connections:

- BUT_IN**: A component with inputs `CLK`, `BUTTON_IN`, and `RESET`, and an output `Q_OUT`.
- CR8CE**: A counter component with inputs `CE` and `CLR`, and an output `Q[7:0]`.
- Connections**:
 - `CLK` is connected to a `CLK0` signal.
 - `BUTTON_IN` is connected to the `BUT` input of **BUT_IN**.
 - `RESET` is connected to the `RESET` input of **BUT_IN** and the `CLR` input of **CR8CE**.
 - `Q_OUT` of **BUT_IN is connected to the `CE` input of **CR8CE**.**
 - `Q[7:0]` of **CR8CE** is connected to a `CNT_OUT[7:0]` signal.

The left-hand panels show the project hierarchy and process options:

- Sources**: Lists the project files, including `Main_module (Main_module.sch)` and `1.ucf`.
- Processes**: Shows the "Select Options" dialog for the "Generate Programming File" process, with options for selecting the entire branch or line segment.

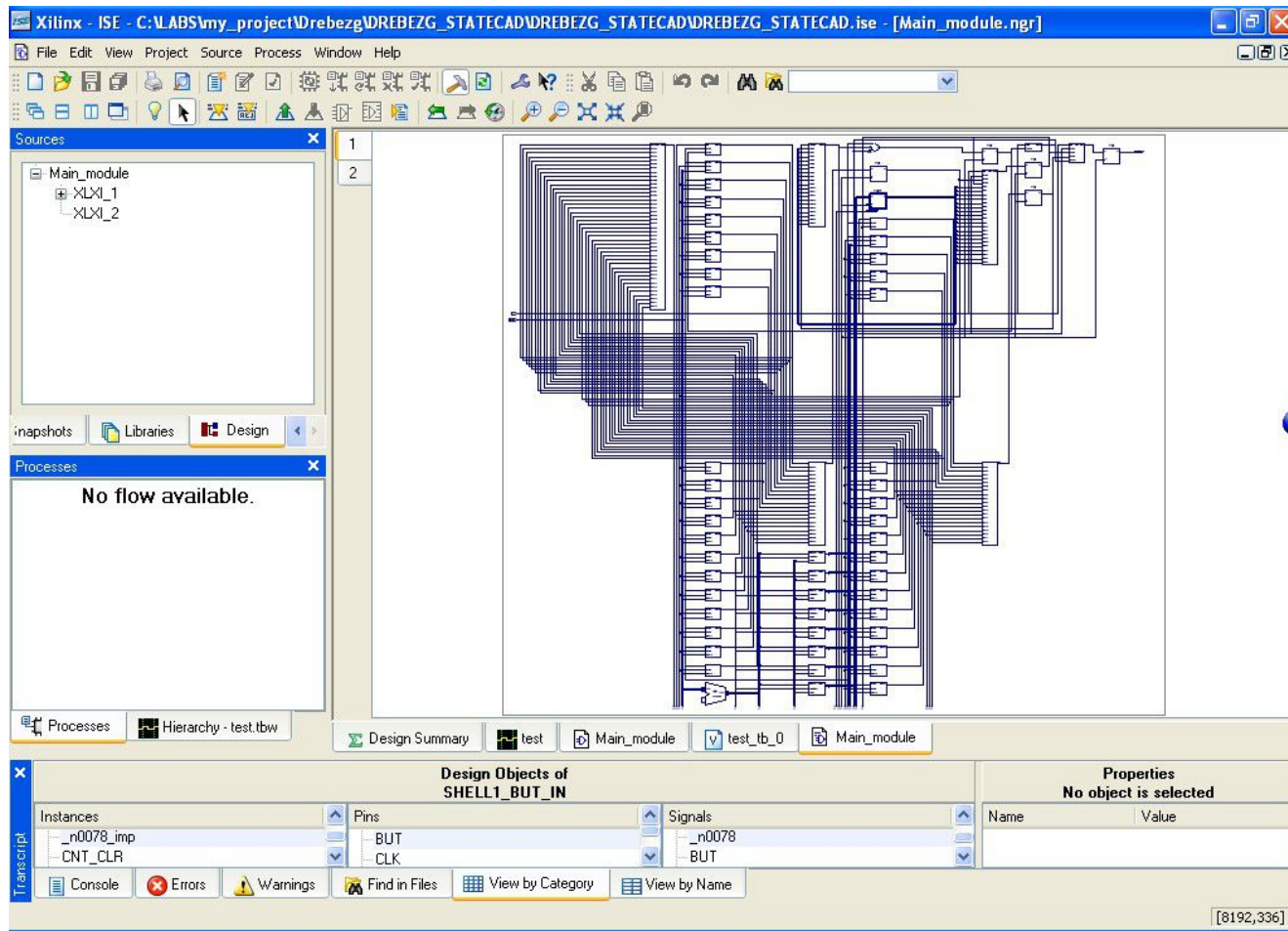
The bottom status bar indicates the following process results:

- Process "Generate Programming File" completed successfully
- Process "Configure Device (iMPACT)" completed successfully

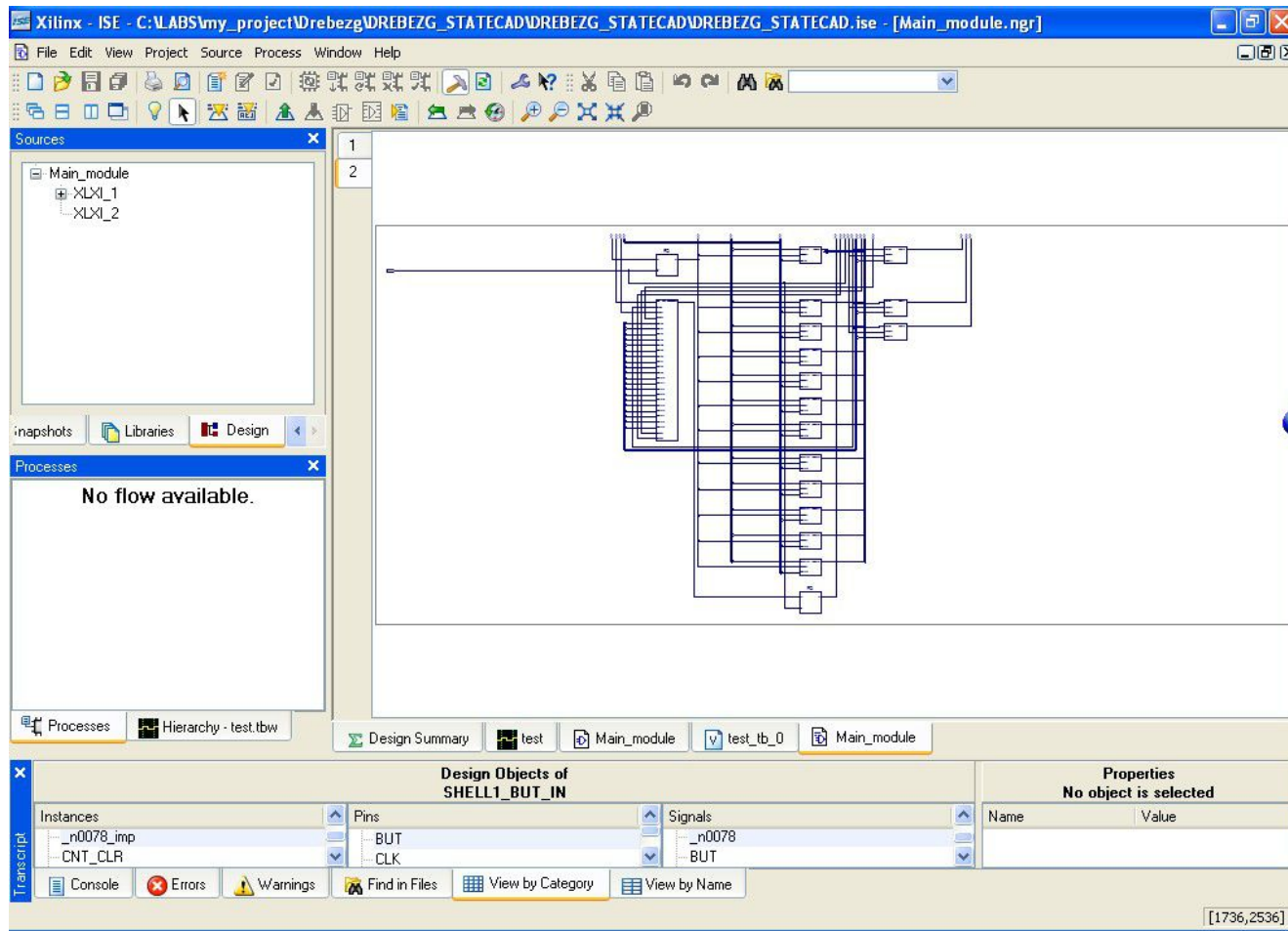
The bottom-most window shows the "Transcript" tab with the following content:

```
Process "Generate Programming File" completed successfully
Process "Configure Device (iMPACT)" completed successfully
```

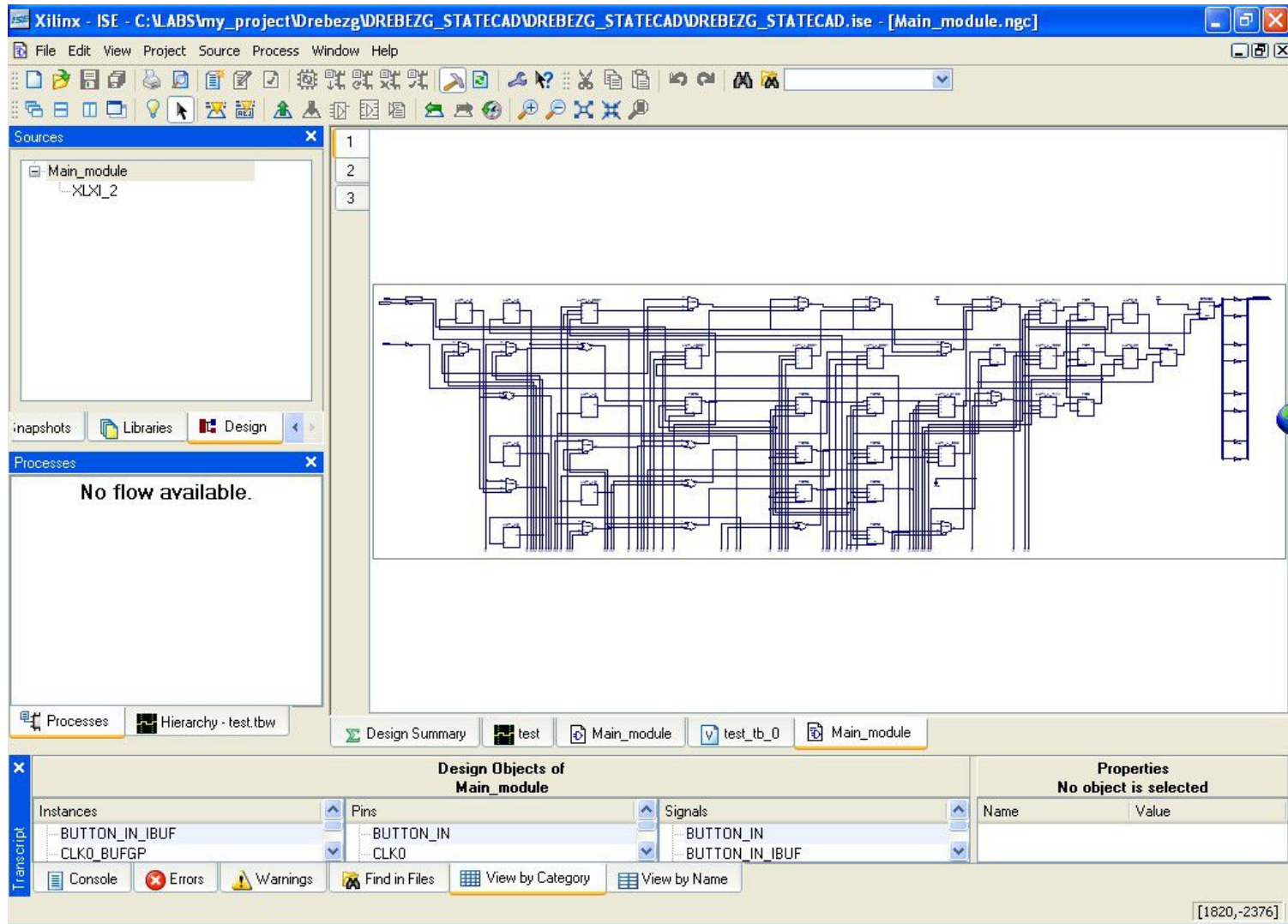
Основная RTL схема устройства подавления дребезга



Основная RTL схема устройства подавления дребезга (2-я часть)



Основная технологическая схема устройства подавления дребезга



Общий отчет по проектированию схемы устранения дребезга (поведенческий вариант)

The screenshot displays the Xilinx ISE Design Summary window for a project named 'DREBEZG_STATECAD'. The window is divided into several panes. The 'Sources' pane on the left shows the project hierarchy, including 'Main_module (Main_module.sch)', 'XLXI_1 - BUT_IN - BEHAVIOR (BU)', and '1.ucf (1.ucf)'. The 'Processes' pane shows the synthesis and implementation steps, with 'Synthesize - XST' and 'Implement Design' marked as completed. The 'Design Summary' pane on the right contains two tables: 'Device Utilization Summary' and 'Performance Summary'.

Device Utilization Summary

Logic Utilization	Used	Available	Utilization	Note(s)
Number of Slice Flip Flops	35	3,840	1%	
Number of 4 input LUTs	21	3,840	1%	
Logic Distribution				
Number of occupied Slices	30	1,920	1%	
Number of Slices containing only related logic	30	30	100%	
Number of Slices containing unrelated logic	0	30	0%	
Total Number 4 input LUTs	40	3,840	1%	
Number used as logic	21			
Number used as a route-thru	19			
Number of bonded IOBs	11	173	6%	
Number of GCLKs	1	8	12%	
Total equivalent gate count for design	553			
Additional JTAG gate count for IOBs	528			

Performance Summary

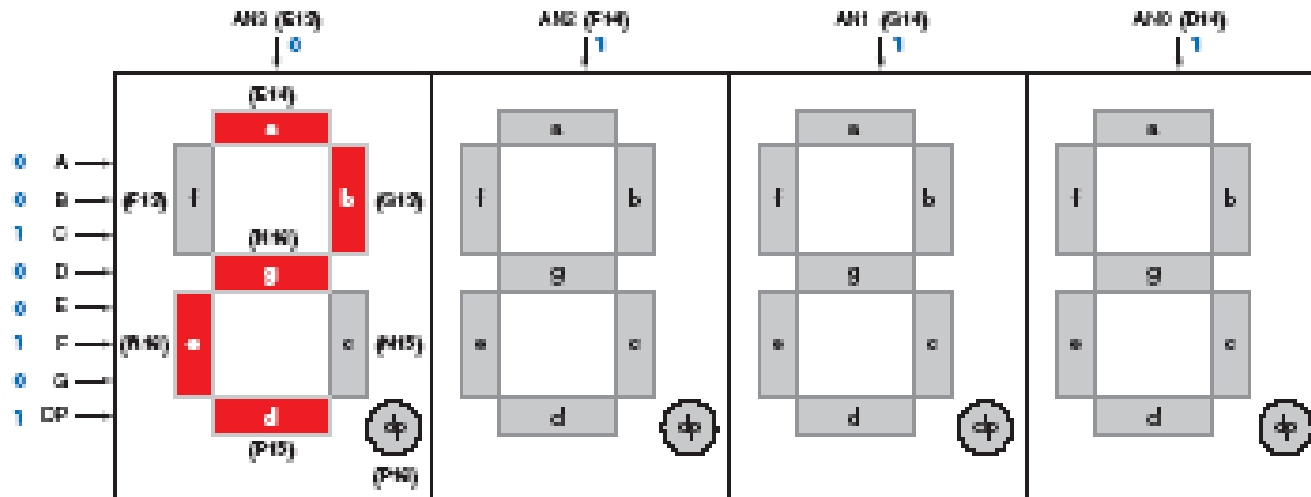
Final Timing Score:		Pinout Data:	
0		Pinout Report	
Routing Results:		Clock Data:	
All Signals Completely Routed		Clock Report	
Timing Constraints:			
All Constraints Met			

The 'Transcript' pane at the bottom shows the message: 'Started : "Launching Technology Schematic Viewer for Main_module.ngc".'

▫ Пример 3

- Устройство управления 7-сегментными индикаторами
 - (Структурный вариант)

Описание декодера 7-сегментного индикатора



```

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

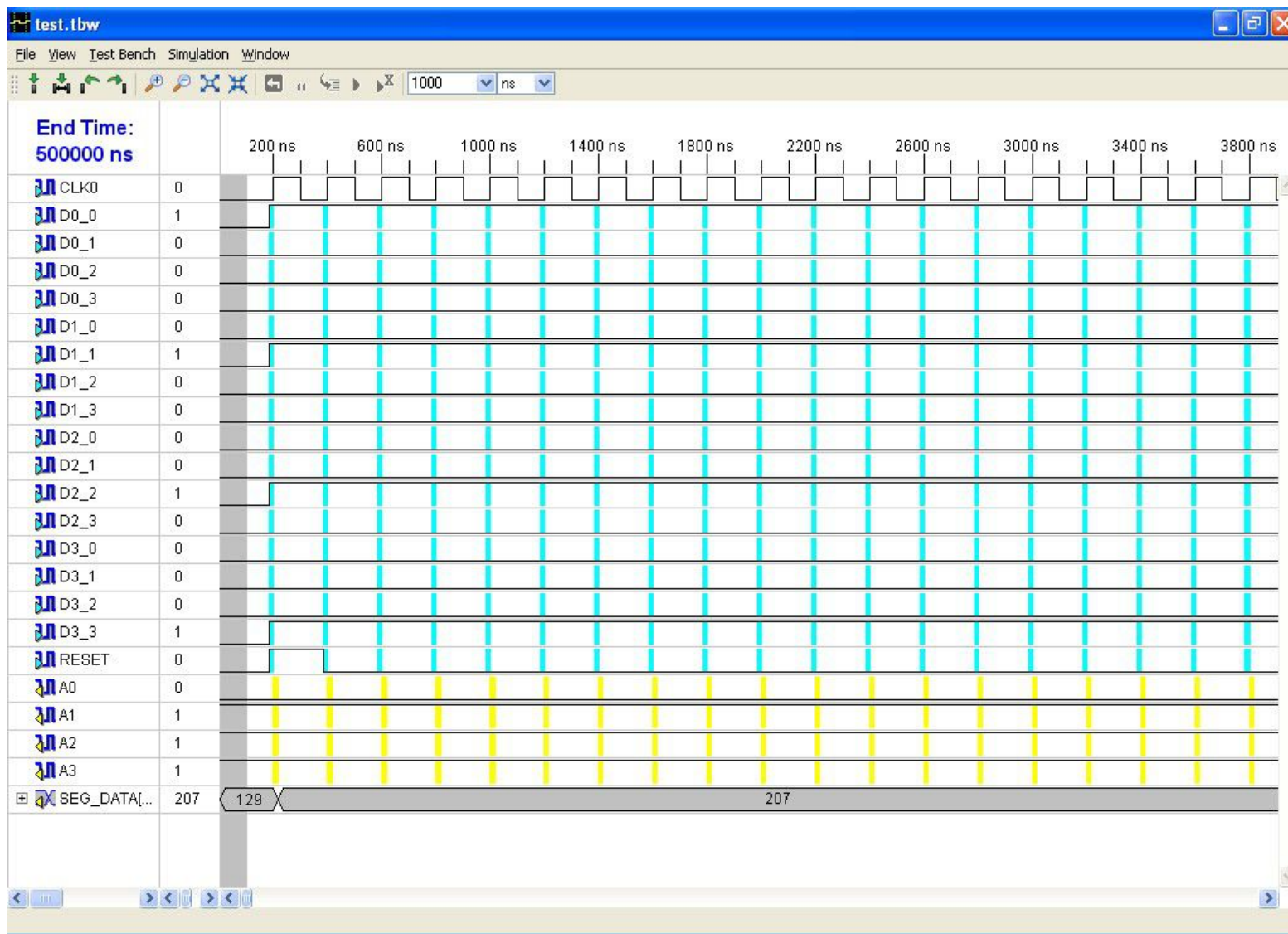
entity led_decode is
    Port (   DH :          in   STD_LOGIC_VECTOR (3 downto 0);
            SEG_DATA : out   STD_LOGIC_VECTOR (7 downto 0));
end led_decode;
    
```

```

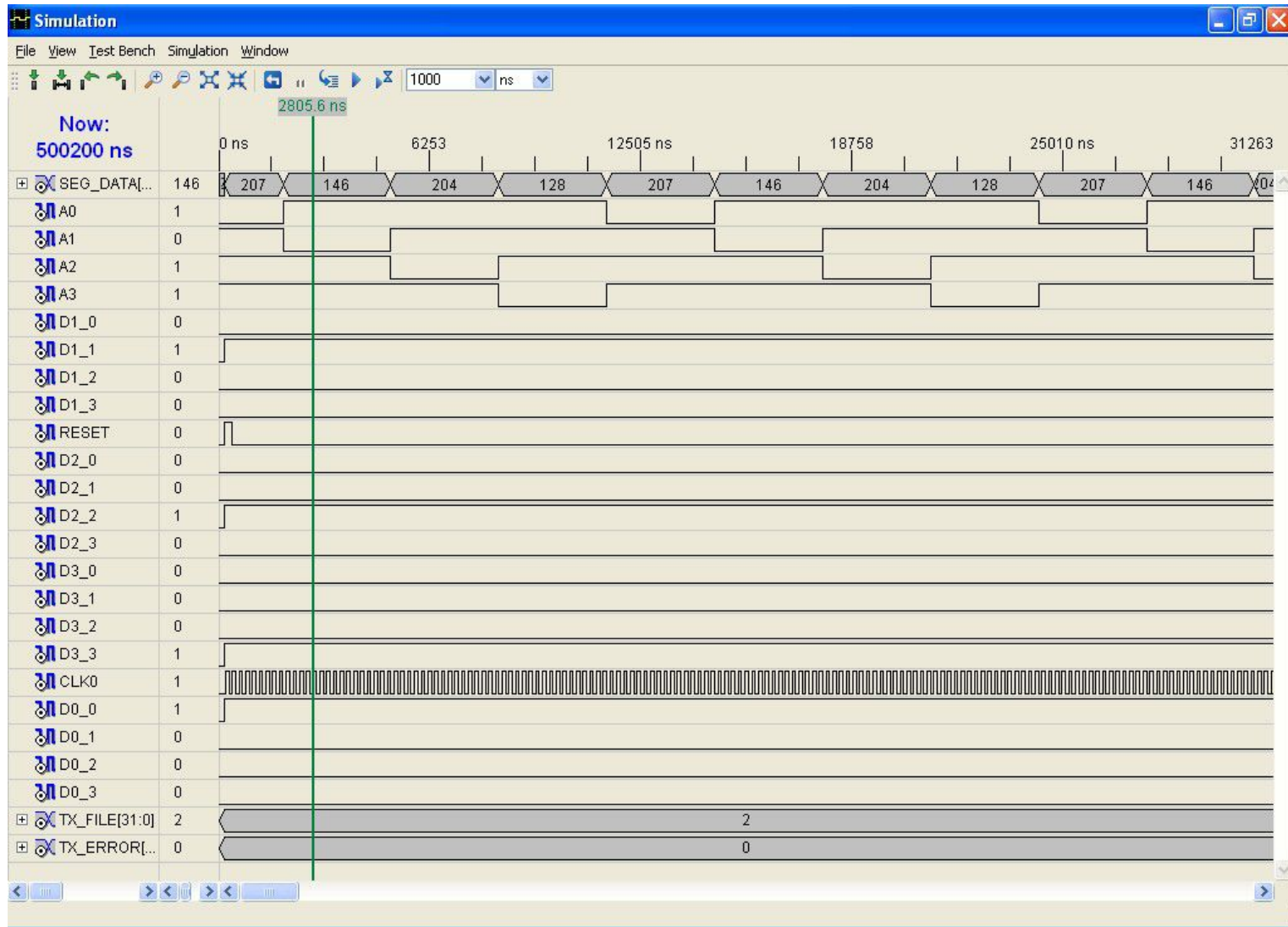
architecture Behavioral of led_decode is
begin
  process (DH)
  begin
    case DH is
      when "0000" => SEG_DATA <= "10000001";
      when "0001" => SEG_DATA <= "11001111";
      when "0010" => SEG_DATA <= "10010010";
      when "0011" => SEG_DATA <= "10000110";
      when "0100" => SEG_DATA <= "11001100";
      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;

```

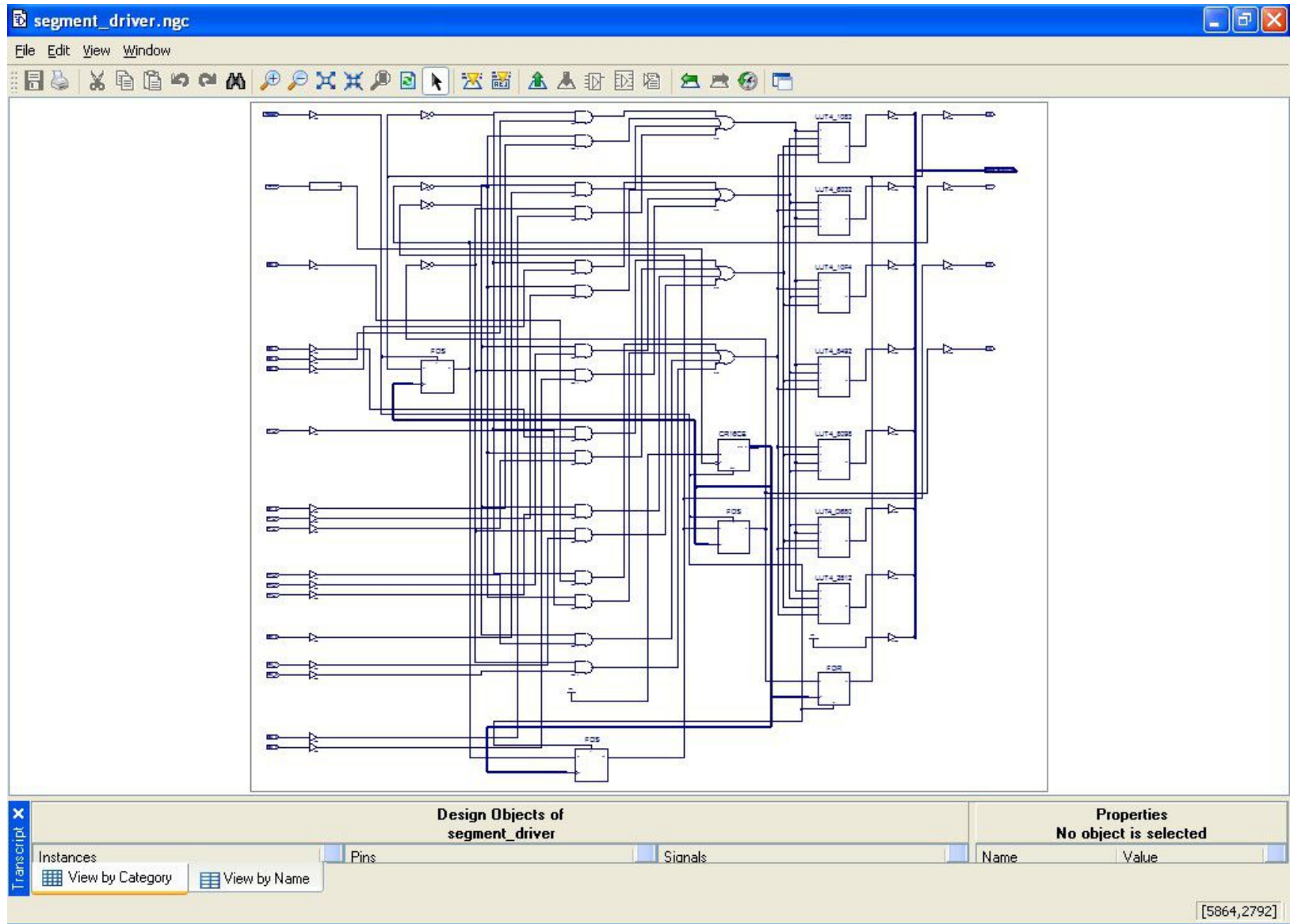

□ Создание тестового воздействия



Моделирование схемы управления 7-сегментными индикаторами



Технологическая схема



Карта Карно (LUT4)

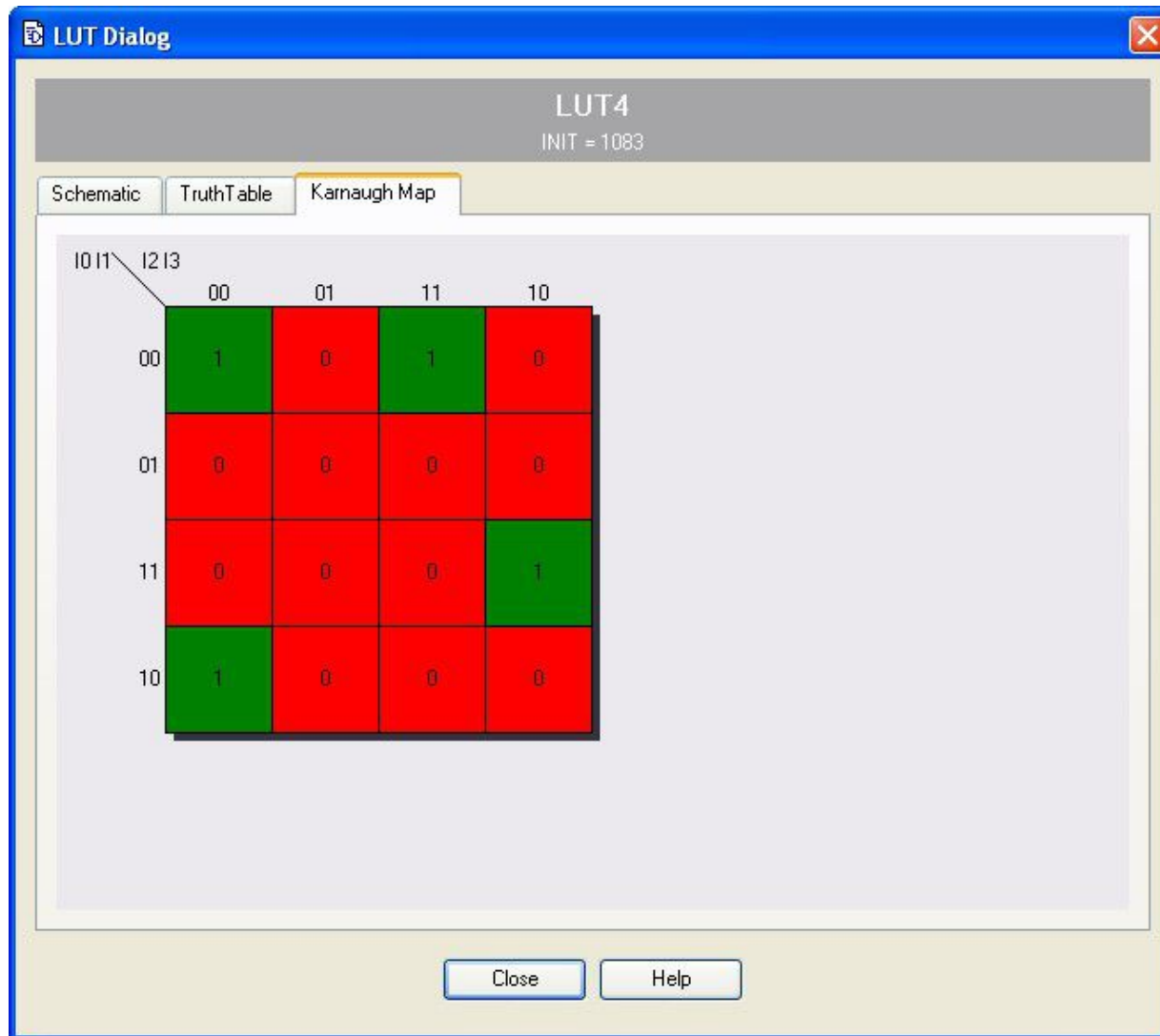


Таблица истинности (LUT4)

LUT Dialog

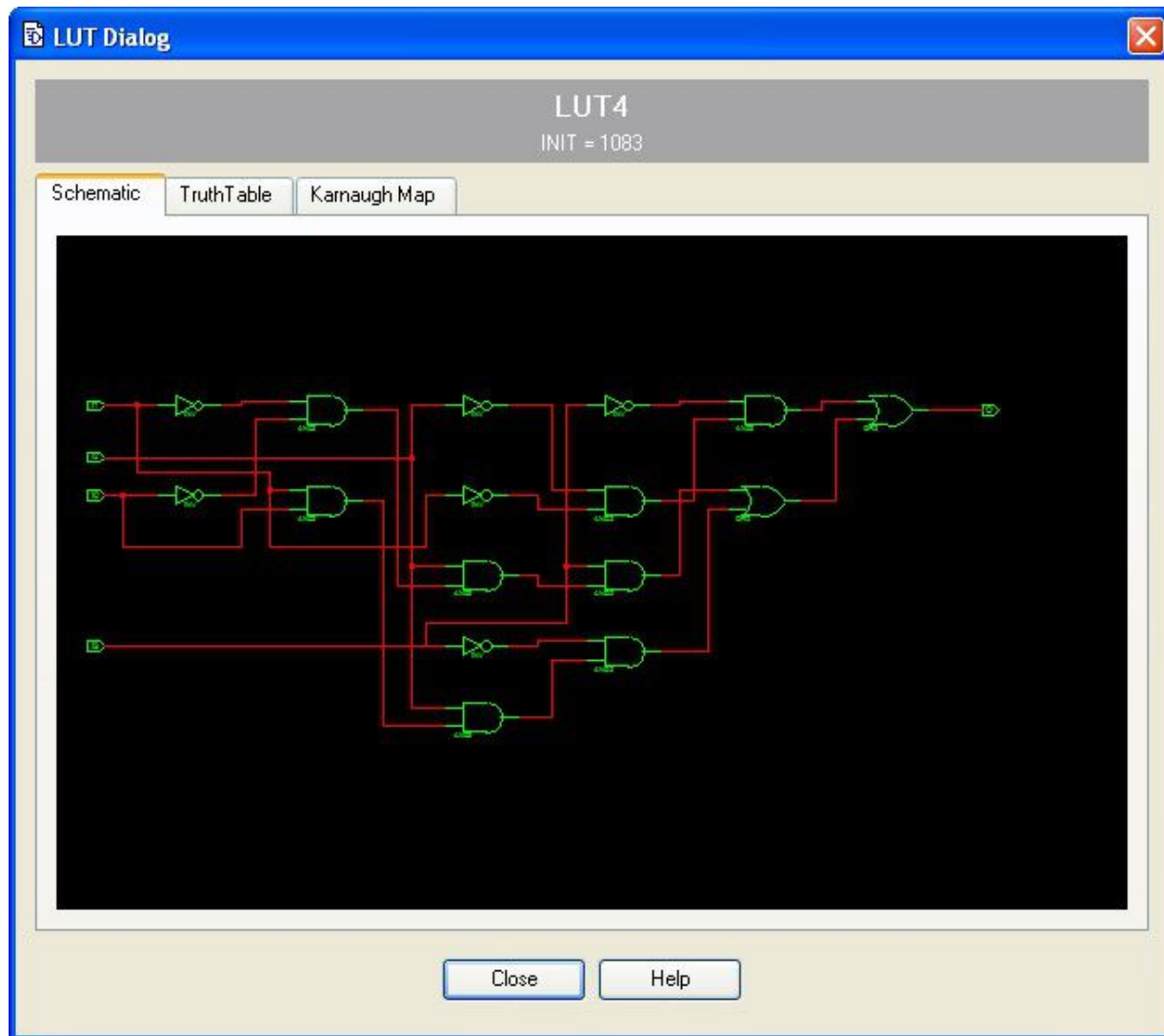
LUT4
INIT = 1083

Schematic TruthTable Karnaugh Map

I3	I2	I1	I0	O
0	0	0	0	1
0	0	0	1	1
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	1
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0

Close Help

□ Логическая схема (LUT4)



Общий отчет по проектированию схемы

The screenshot displays the Xilinx ISE Design Summary window for a project named '7SEGMENT'. The window is divided into several panes. The 'Sources' pane on the left shows the project hierarchy, including the 'segment_driver' source. The 'Processes' pane below it lists various design steps, with 'View Design Summary' currently selected. The main area of the window contains two summary tables. The first table, '7SEGMENT Project Status', provides an overview of the project's current state, including the project file, module name, target device, and product version. The second table, 'Device Utilization Summary', details the logic utilization, including the number of slice flip flops, LUTs, and occupied slices, along with the total equivalent gate count for the design. At the bottom of the window, a 'Transcript' pane shows the command 'Started : "Launching Design Summary".' and a status bar indicates 'Готово' (Ready).

7SEGMENT Project Status

Project File:	7Segment.isc	Current State:	Placed and Routed
Module Name:	segment_driver	Errors:	No Errors
Target Device:	xc3s200-4ft256	Warnings:	6 Warnings (3 new, 0 filtered)
Product Version:	ISE, 8.1.03i	Updated:	?? 8. ??? 16:29:36 2007

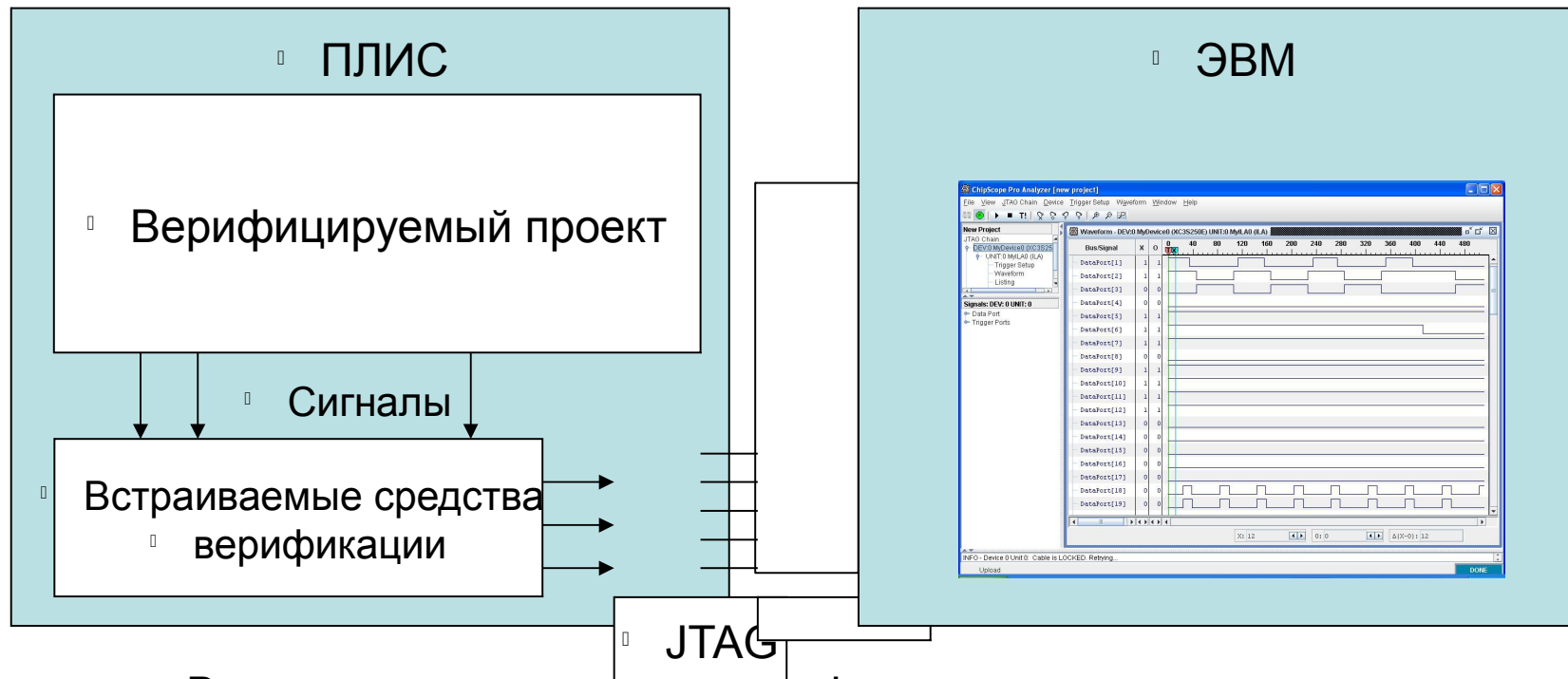
Device Utilization Summary

Logic Utilization	Used	Available	Utilization	Note(s)
Number of Slice Flip Flops	8	3,840	1%	
Number of 4 input LUTs	19	3,840	1%	
Logic Distribution				
Number of occupied Slices	18	1,920	1%	
Number of Slices containing only related logic	18	18	100%	
Number of Slices containing unrelated logic	0	18	0%	
Total Number of 4 input LUTs	19	3,840	1%	
Number of bonded IOBs	30	173	17%	
Number of GCLKs	1	8	12%	
Total equivalent gate count for design	181			
Additional JTAG gate count for IOBs	1,440			

Started : "Launching Design Summary".

Готово

• Верификация проектов на ПЛИС с использованием ChipScope



• Встраиваемые средства верификации:

- Ядро коммутации с JTAG интерфейсом (ICON);
- Встраиваемые логические анализаторы (ILA, VIO, ATC2);
- Генераторы последовательностей (VIO, IBERT);
- Средства верификации встраиваемых систем (IBA/OPB, IBA/PLB);
- Средства динамической реконфигурации (IBERT)

- Верификация проектов на ПЛИС с использованием ChipScope

The screenshot displays the ChipScope Pro Analyzer software interface. The main window is titled "Trigger Setup - DEV:0 MyDevice0 (XC3S250E) UNIT:0 MyILA0 (ILA)". The interface includes a menu bar (File, View, JTAG Chain, Device, Trigger Setup, Waveform, Window, Help) and a toolbar with various icons. On the left, a "New Project" sidebar shows the JTAG Chain structure: DEV:0 MyDevice0 (XC3S250E) > UNIT:0 MyILA0 (ILA) > Trigger Setup. Below this, the "Signals: DEV: 0 UNIT: 0" section lists "Data Port" and "Trigger Ports".

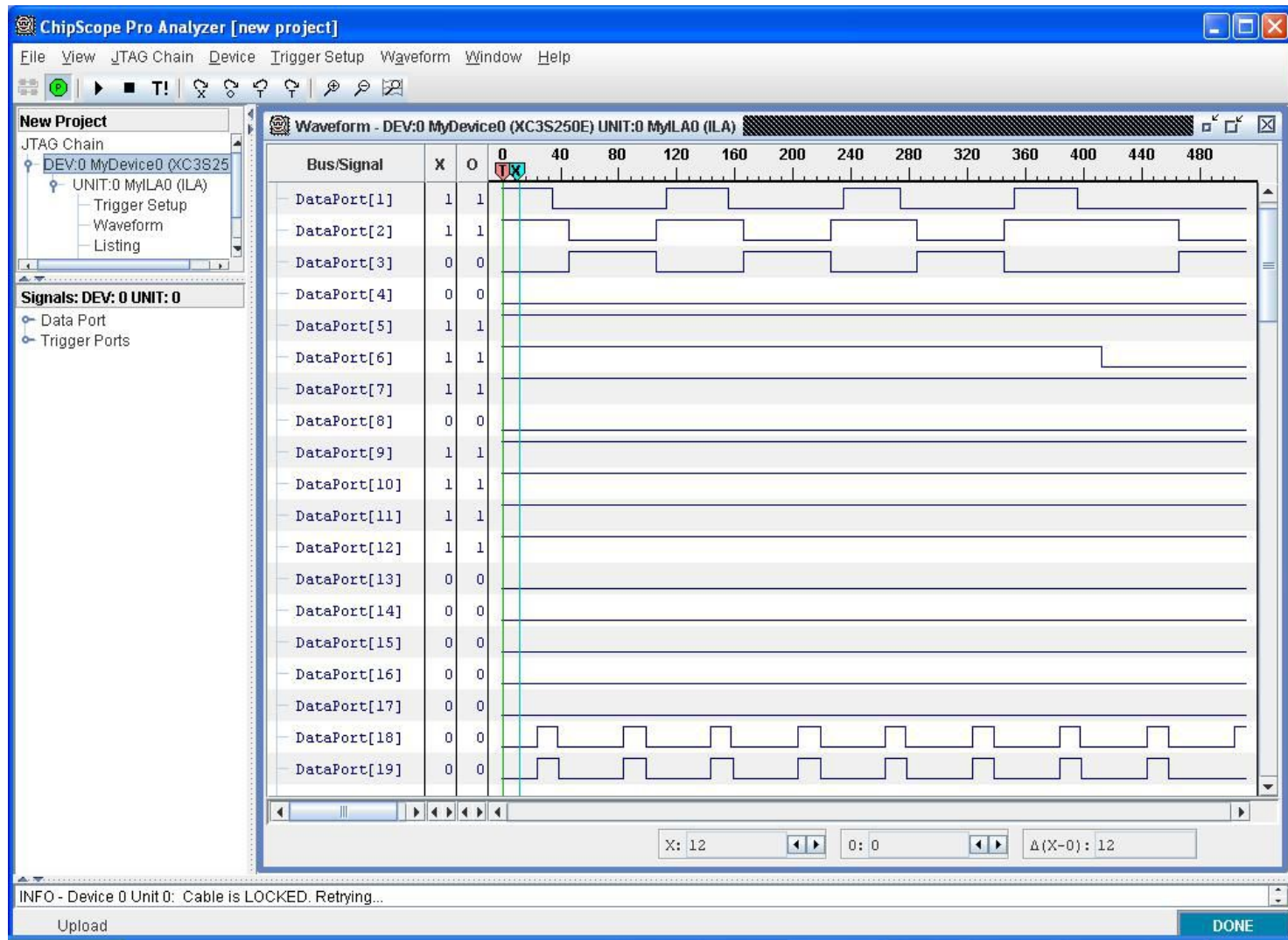
The central "Match" table lists trigger ports and their values. The table has columns: Match Unit, Function, Value, Radix, and Counter. The "Counter" column is set to "disabled".

Match Unit	Function	Value	Radix	Counter
M0:TriggerPort0	==	X_X1XX_XX0X_XXXX_XXXX_XXXX_XXXX_XX	Bin	disabled
TriggerPort0[96]			X	
TriggerPort0[95]			X	
TriggerPort0[94]			1	
TriggerPort0[93]			X	
TriggerPort0[92]			X	
TriggerPort0[91]			X	
TriggerPort0[90]			X	
TriggerPort0[89]			0	
TriggerPort0[88]			X	
TriggerPort0[87]			X	
TriggerPort0[86]			X	
TriggerPort0[85]			X	
TriggerPort0[84]			X	
TriggerPort0[83]			X	
TriggerPort0[82]			X	
TriggerPort0[81]			X	
TriggerPort0[80]			0	
TriggerPort0[79]			X	
TriggerPort0[78]			X	
TriggerPort0[77]			X	

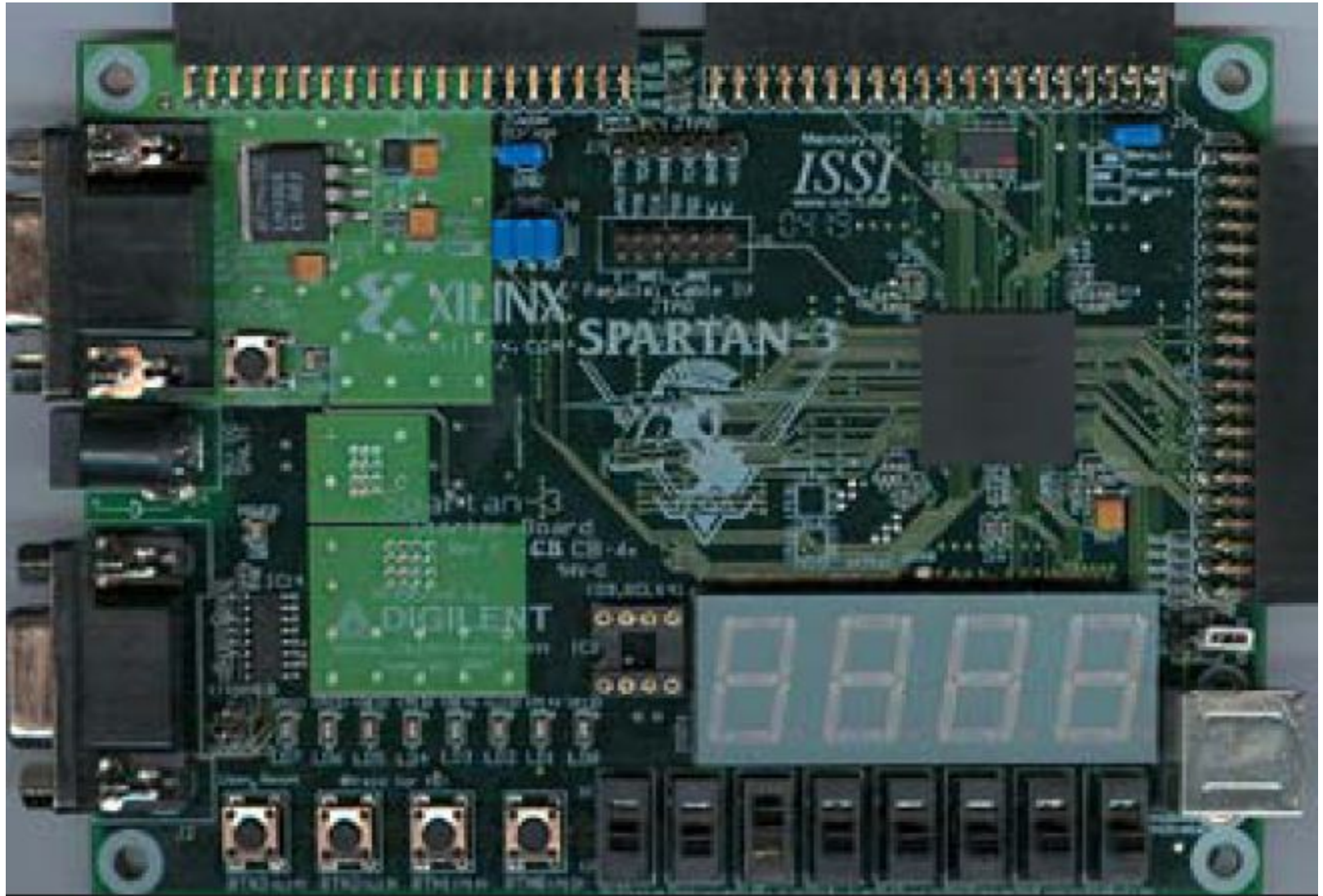
Below the table, the "Trigger" section shows a single trigger condition: "TriggerCondition0" with the equation "M0". The "Capture" section shows settings: Type: Window, Windows: 1, Depth: 512, Position: 0, and Storage Qualification: All Data.

The status bar at the bottom displays the message: "INFO - Device 0 Unit 0: Cable is LOCKED. Retrying...". At the bottom right, there is a "DONE" button.

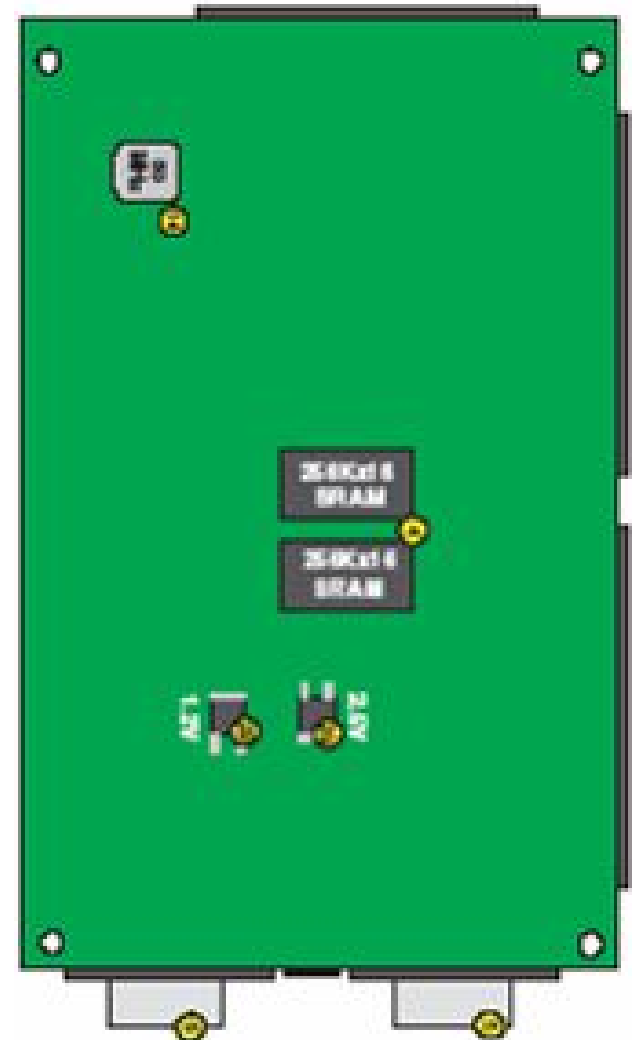
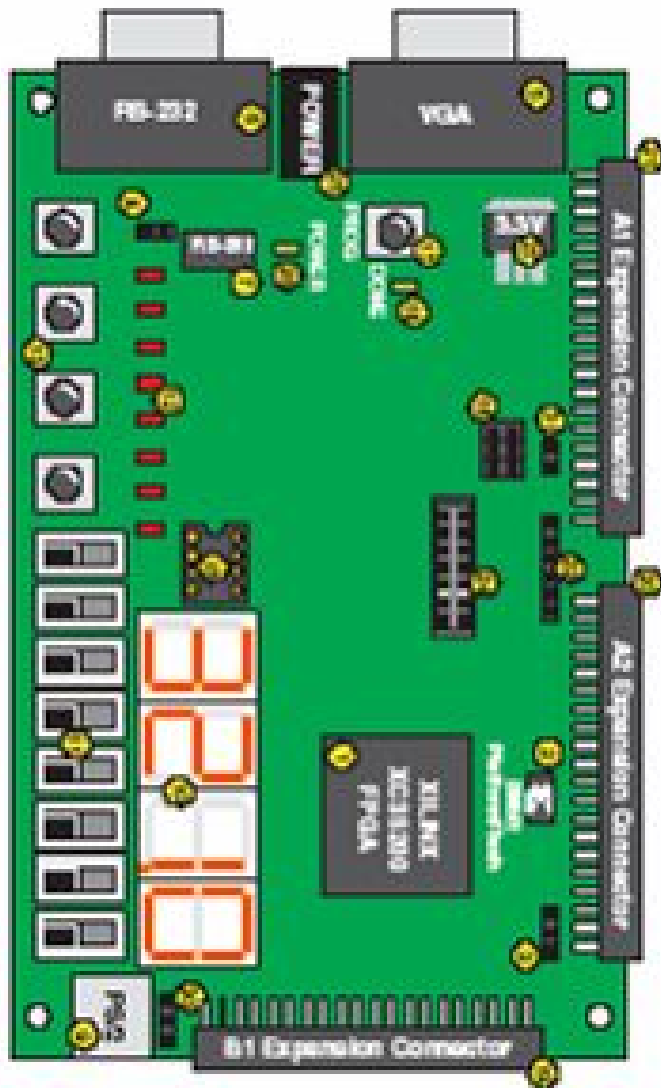
- Верификация проектов на ПЛИС с использованием ChipScope



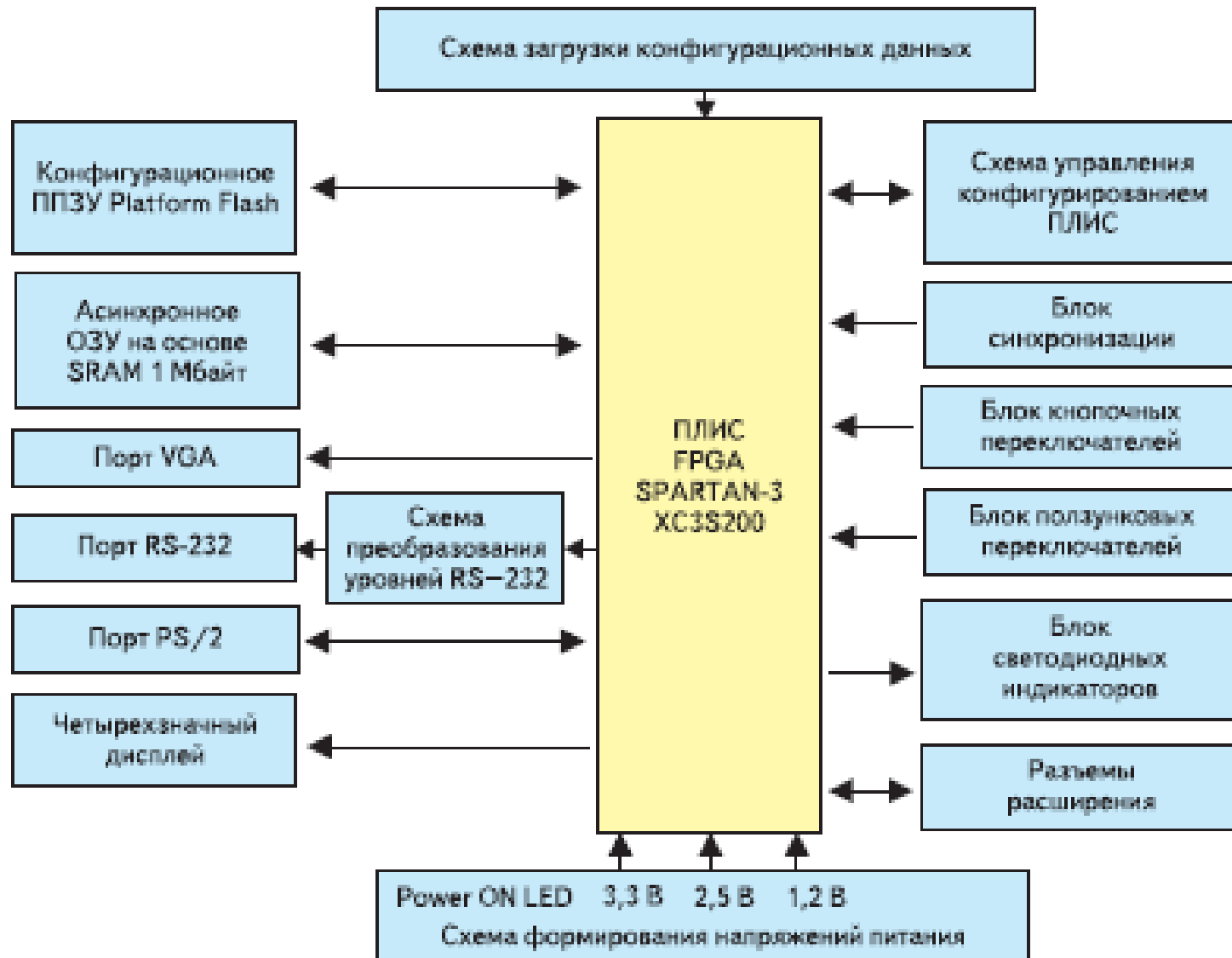
▮ Внешний вид стартового набора разработчика XC3S200



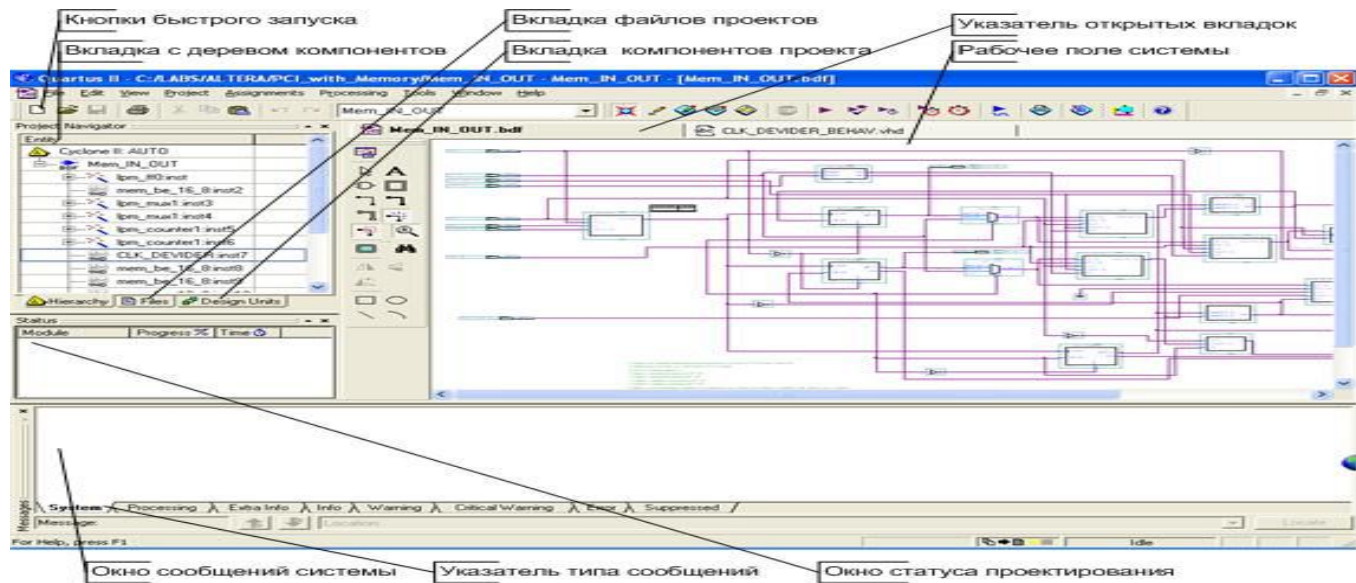
- Лицевая и обратная стороны стартового набора разработчика XC3S200



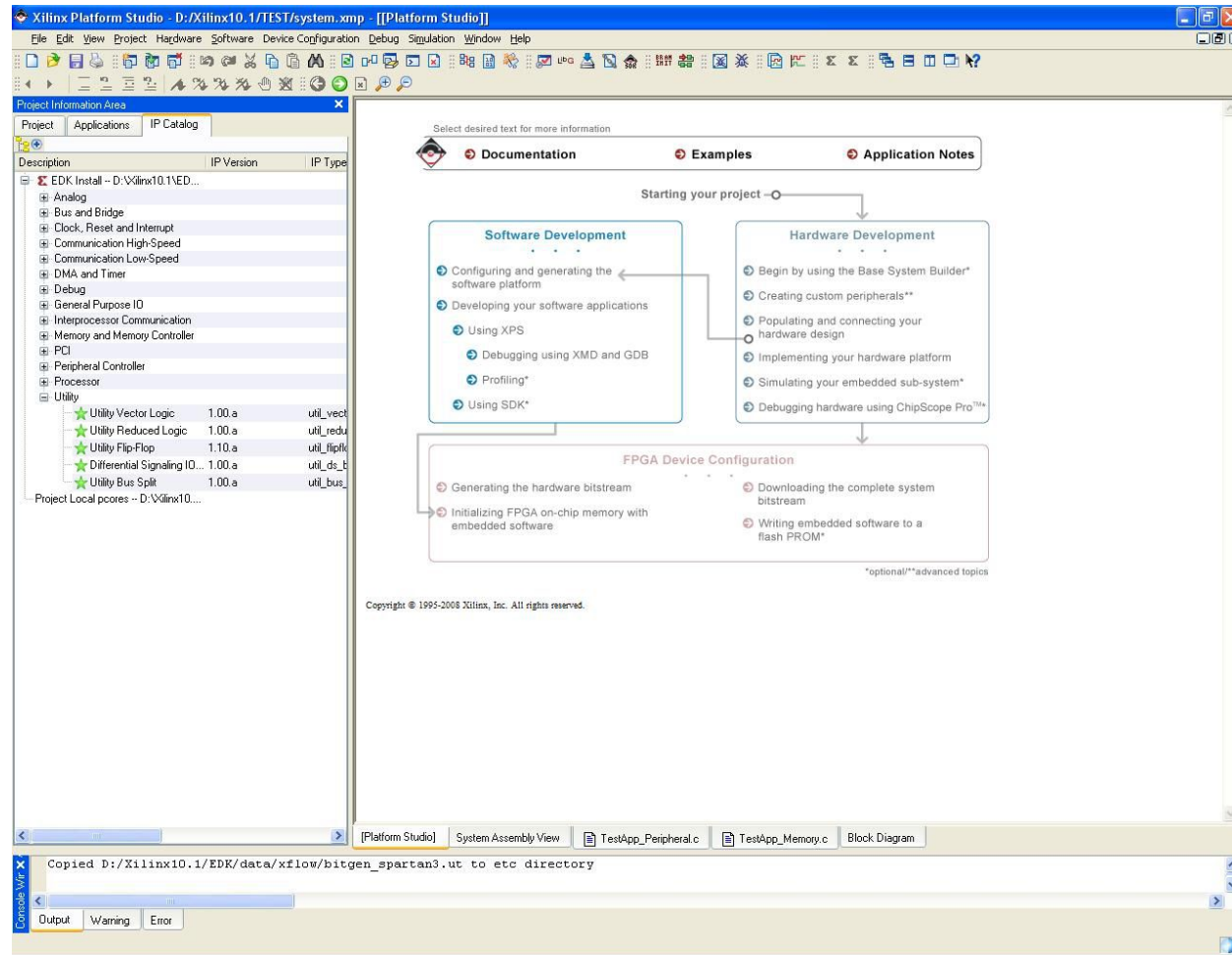
- Состав набора XC3S200



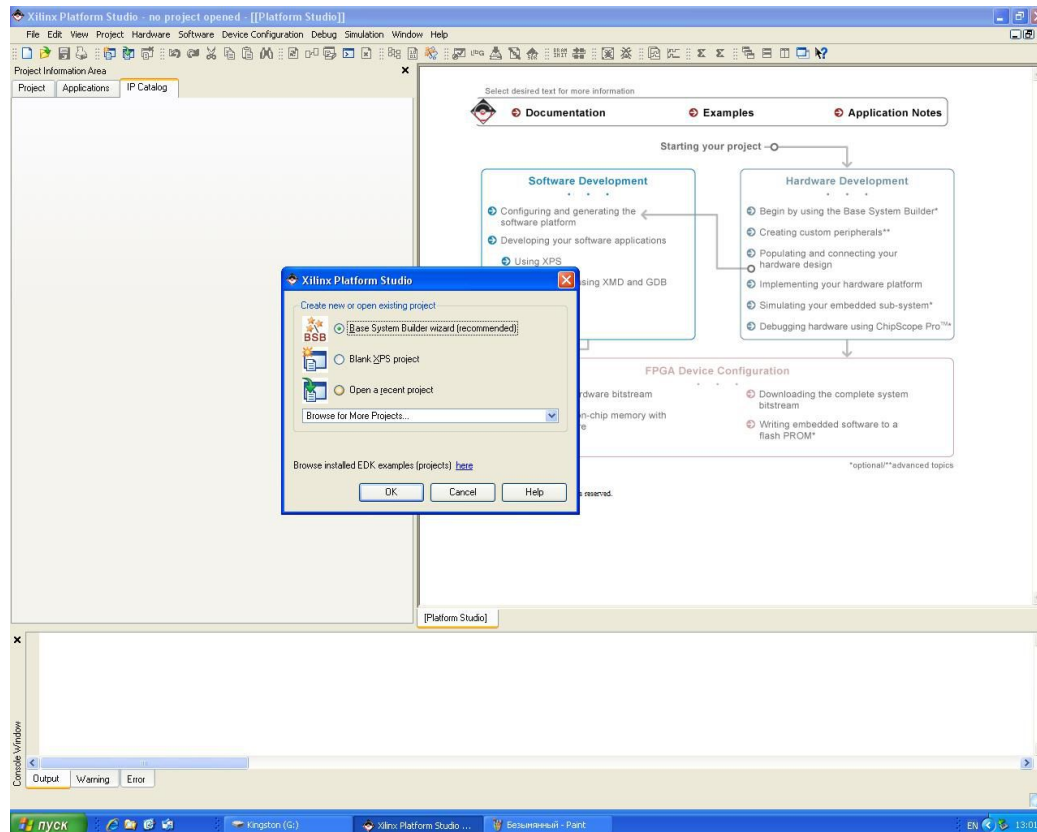
- САПР Altera Quartus II



- Система разработки и отладки SOPC на основе ПЛИС XILINX: EDK



- Система разработки и отладки SOPC на основе ПЛИС XILINX: EDK



- Система разработки и отладки SOPC на основе ПЛИС XILINX: EDK

Base System Builder - Select Board

Select a target development board:

Select board

☒ I would like to create a system for the following development board

Board vendor: Xilinx

Board name: Spartan-3 Starter Board

Board revision: E

Note: Visit the vendor website for additional board support materials.

[Vendor's Website](#) [Contact Info](#)

[Download Third Party Board Definition Files](#)

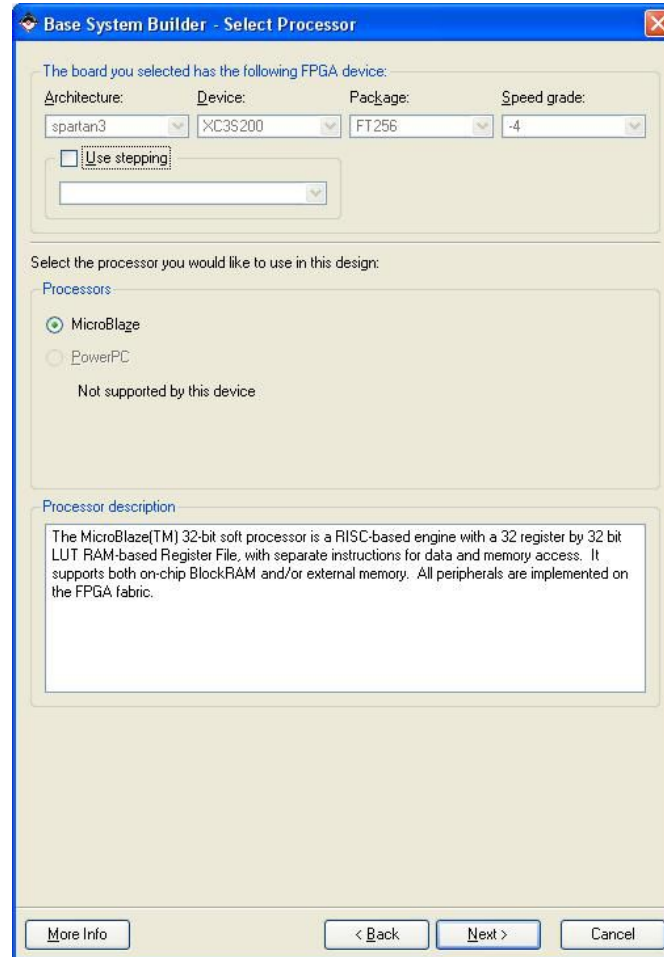
☐ I would like to create a system for a custom board

Board description

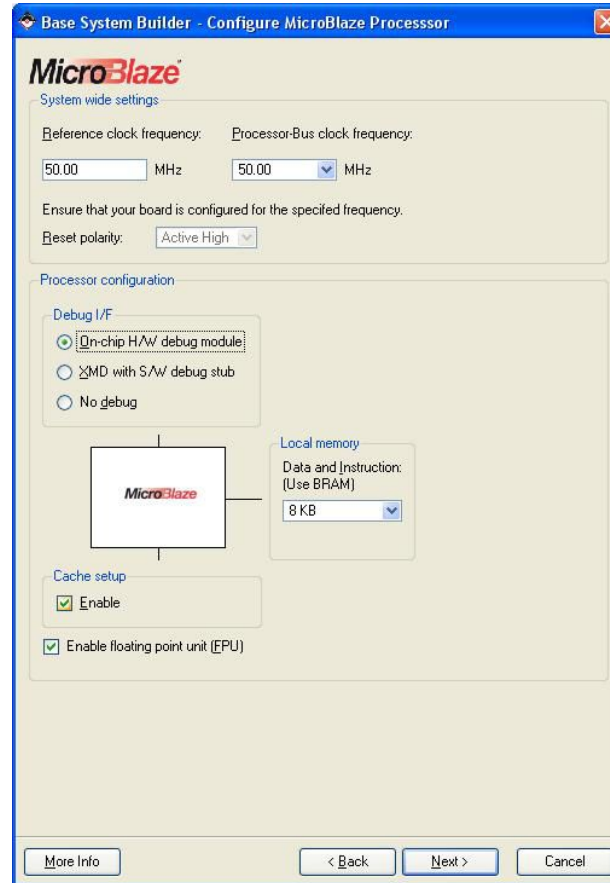
Spartan-3 Starter Kit Board utilizes Xilinx Spartan-3 XC3S200-4FT256 device. The board includes 1 RS232 serial port, 2 256kx16 fast SRAM, 8 DIP switches, 4 push buttons, four digital 7 segment LEDs, 8 LEDs, 1 VGA port, 1 PS/2 port. Push button 1 is used as system reset. 2 SRAMs are combined to form a 32 bit data bus.

More Info < Back Next > Cancel

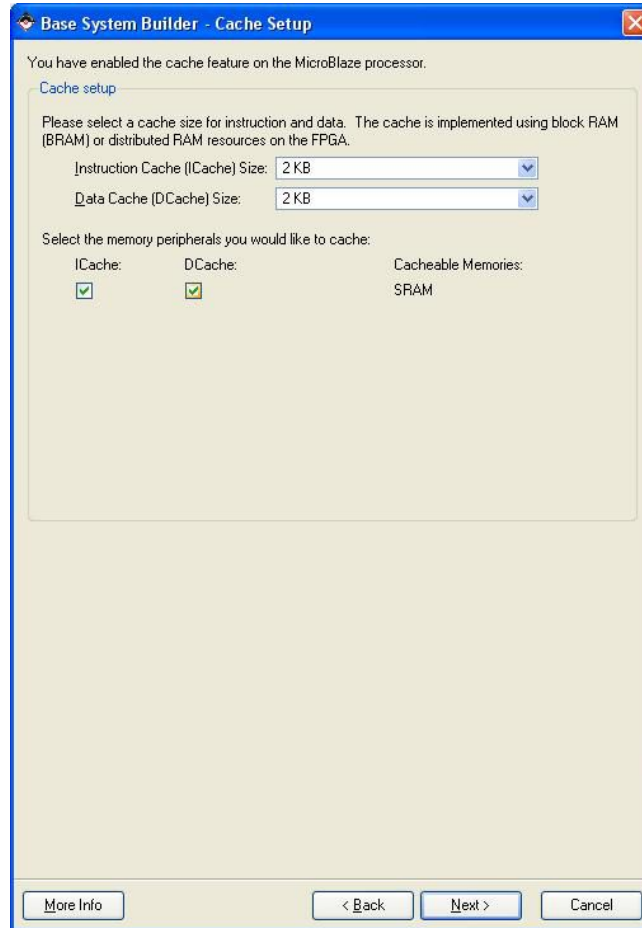
- Система разработки и отладки SOPC на основе ПЛИС XILINX: EDK



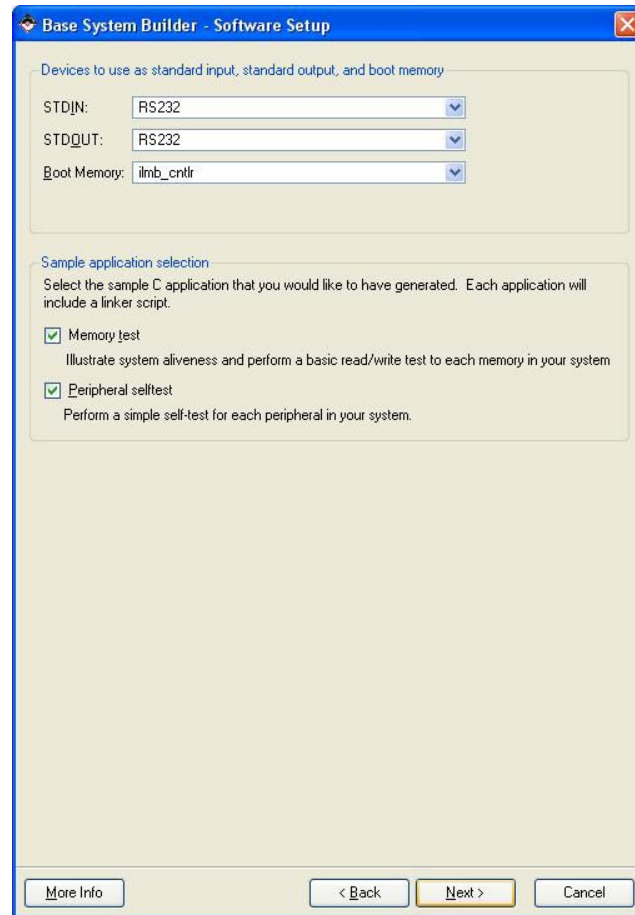
- Система разработки и отладки SOPC на основе ПЛИС XILINX: EDK



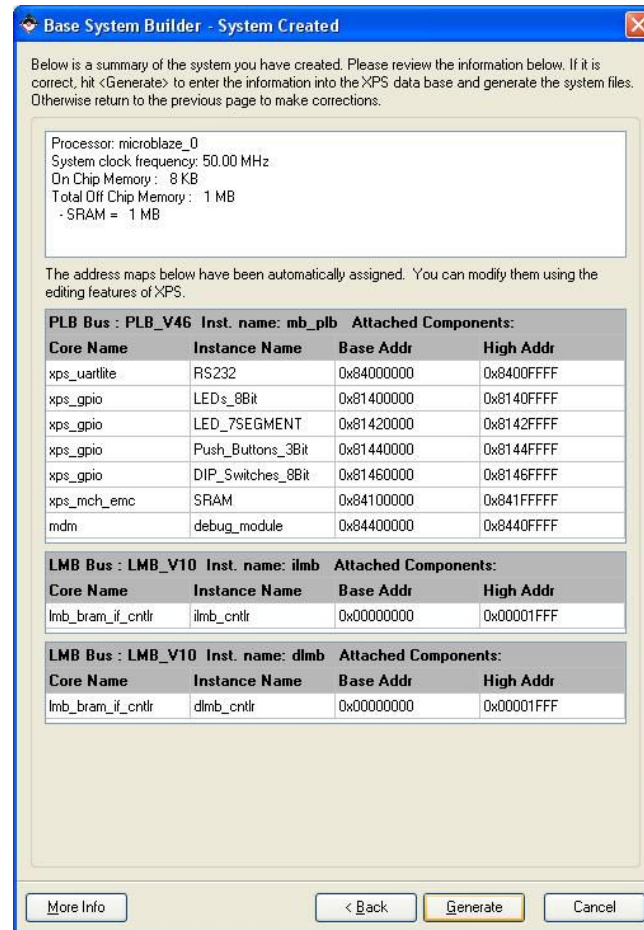
- Система разработки и отладки SOPC на основе ПЛИС XILINX: EDK



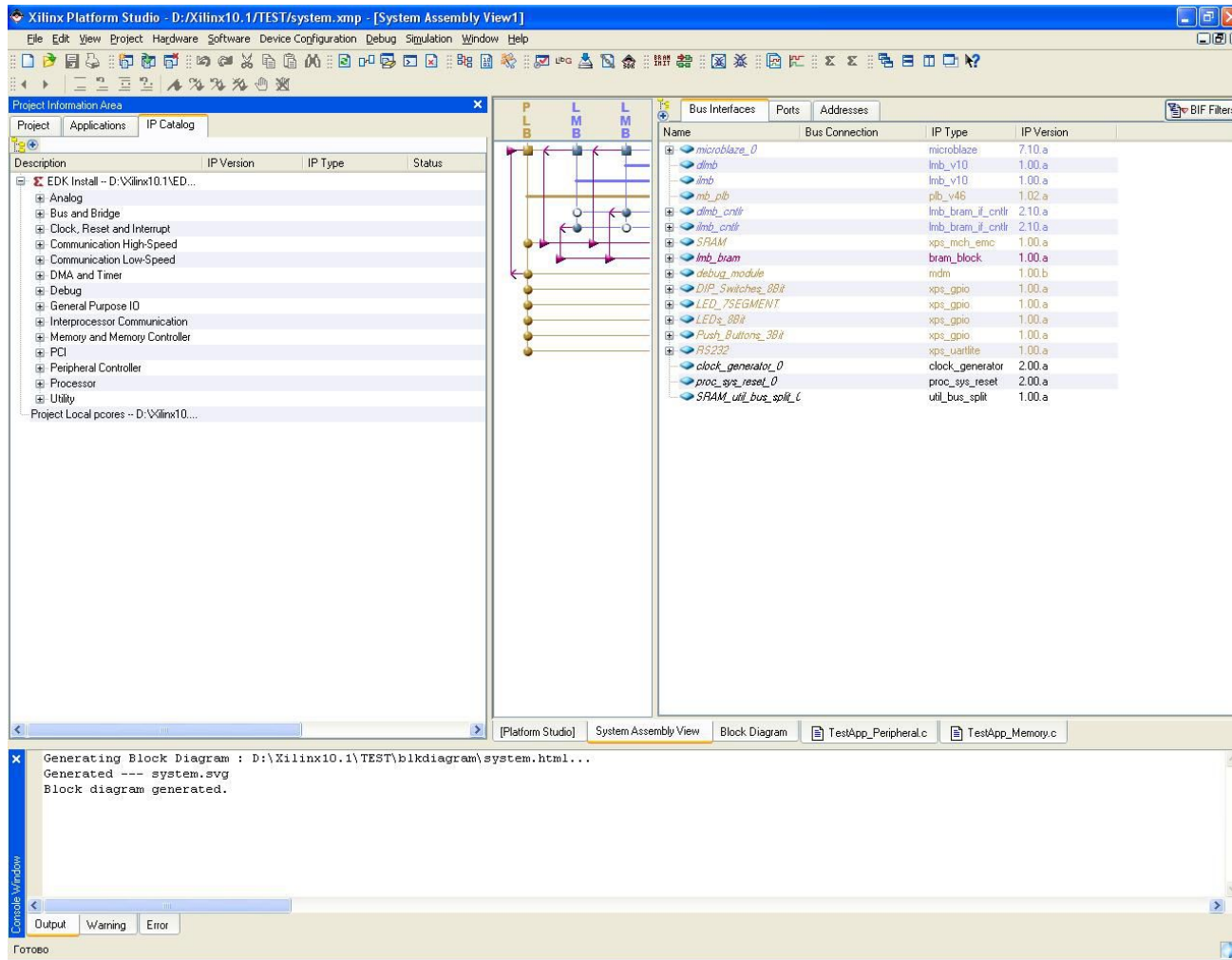
- Система разработки и отладки SOPC на основе ПЛИС XILINX: EDK



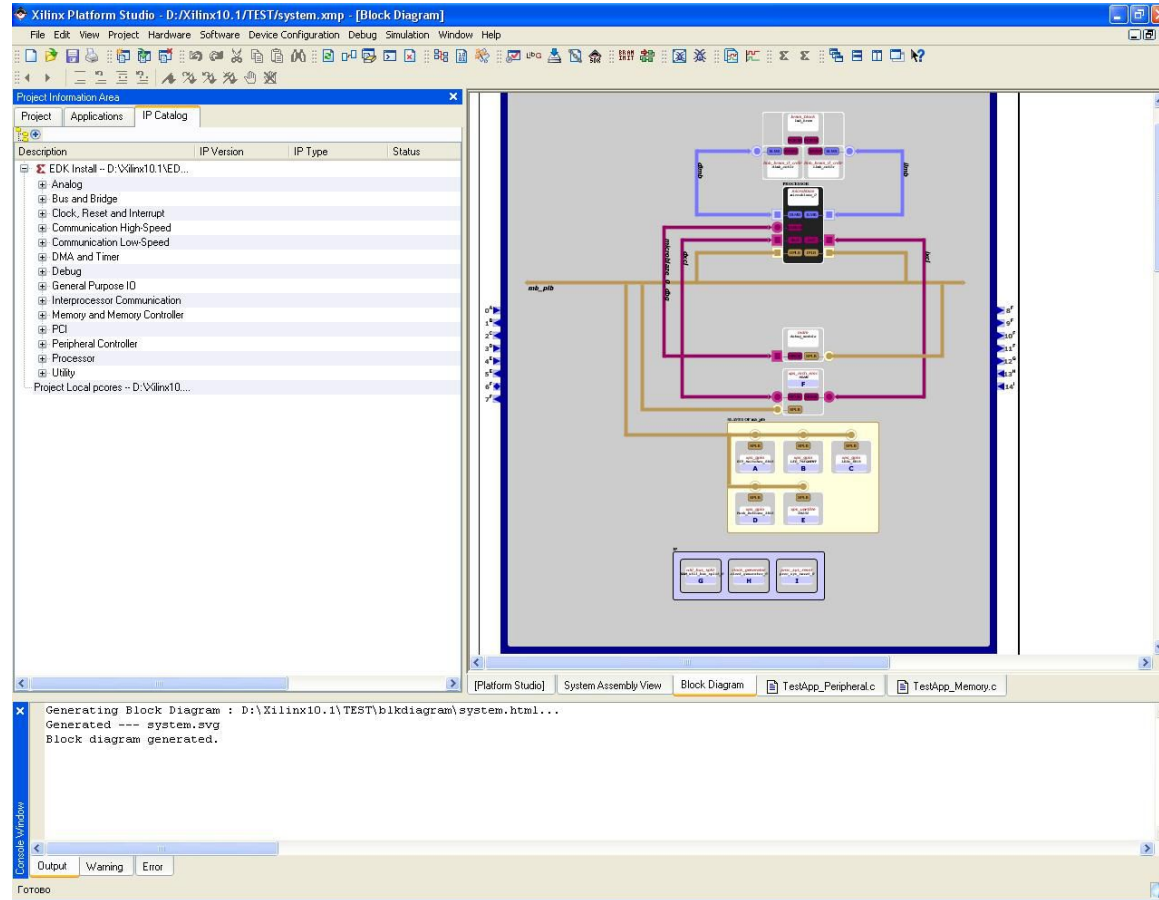
- Система разработки и отладки SOPC на основе ПЛИС XILINX: EDK



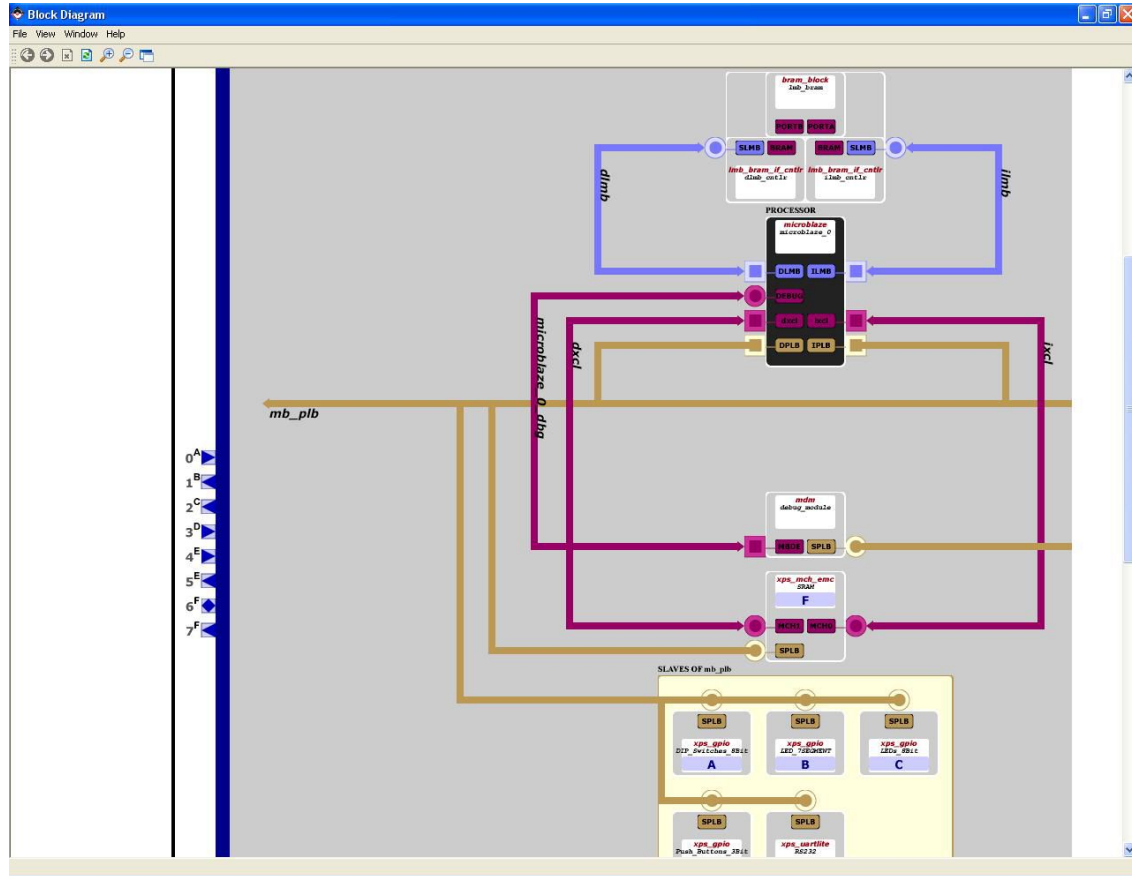
- Система разработки и отладки SOPC на основе ПЛИС XILINX: EDK



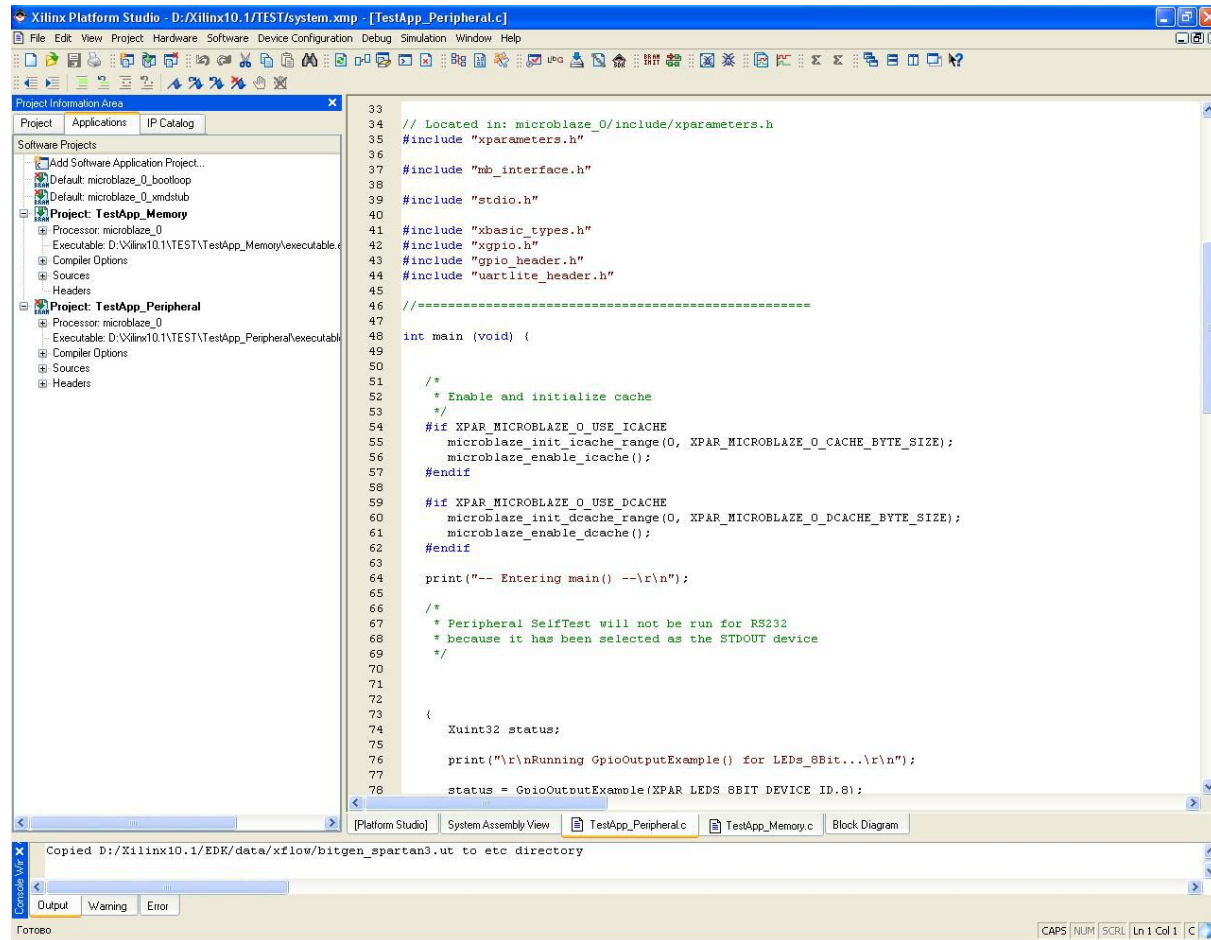
- Система разработки и отладки SOPC на основе ПЛИС XILINX: EDK



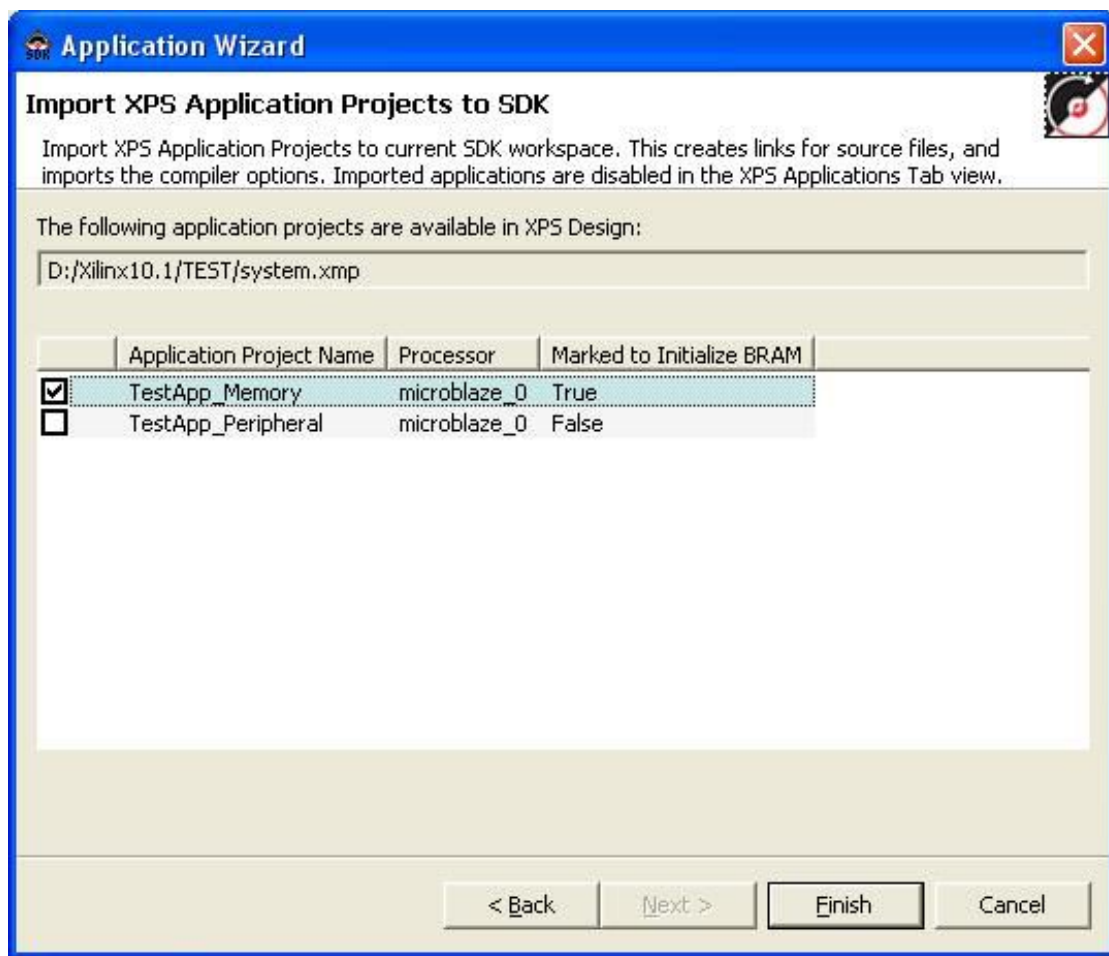
- Система разработки и отладки SOPC на основе ПЛИС XILINX: EDK



- Система разработки и отладки SOPC на основе ПЛИС XILINX: EDK



- Система разработки и отладки SOPC на основе ПЛИС XILINX: EDK



- Система разработки и отладки SOPC на основе ПЛИС XILINX: EDK

