Кафедра промышленной электроники (ПрЭ)

Программные средства моделирования логических БИС (поддерживающие языки Verilog, VHDL)

Выполнил: Студент гр. 368-М1

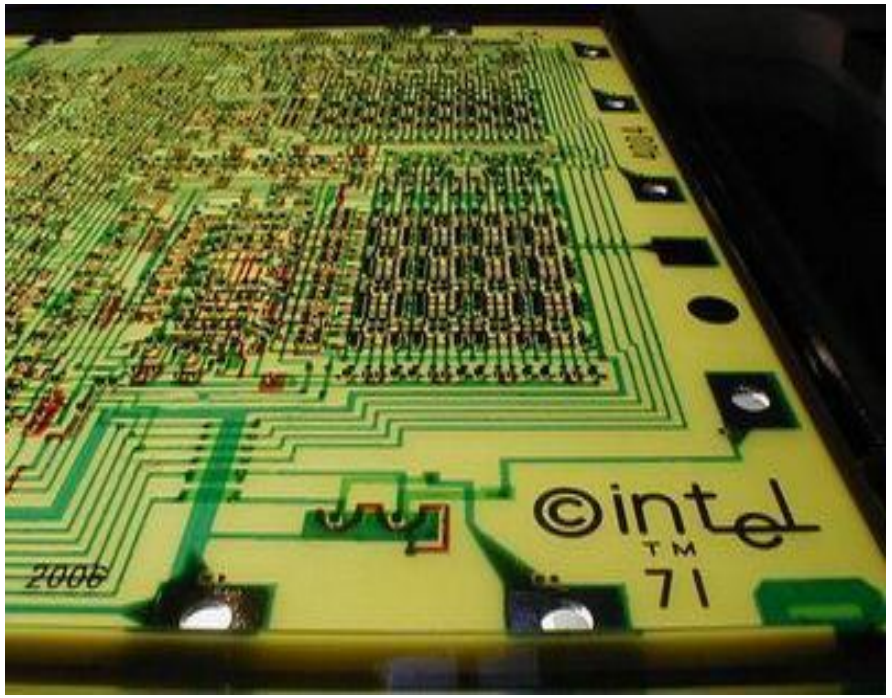
Лункин Дмитрий Сергеевич

Томск 2019

Определение СБИС

Сверхбольшая интегральная схема (СБИС) – электронная схема, изготовленная на полупроводниковом кристалле, помещенная в неразборный корпус.

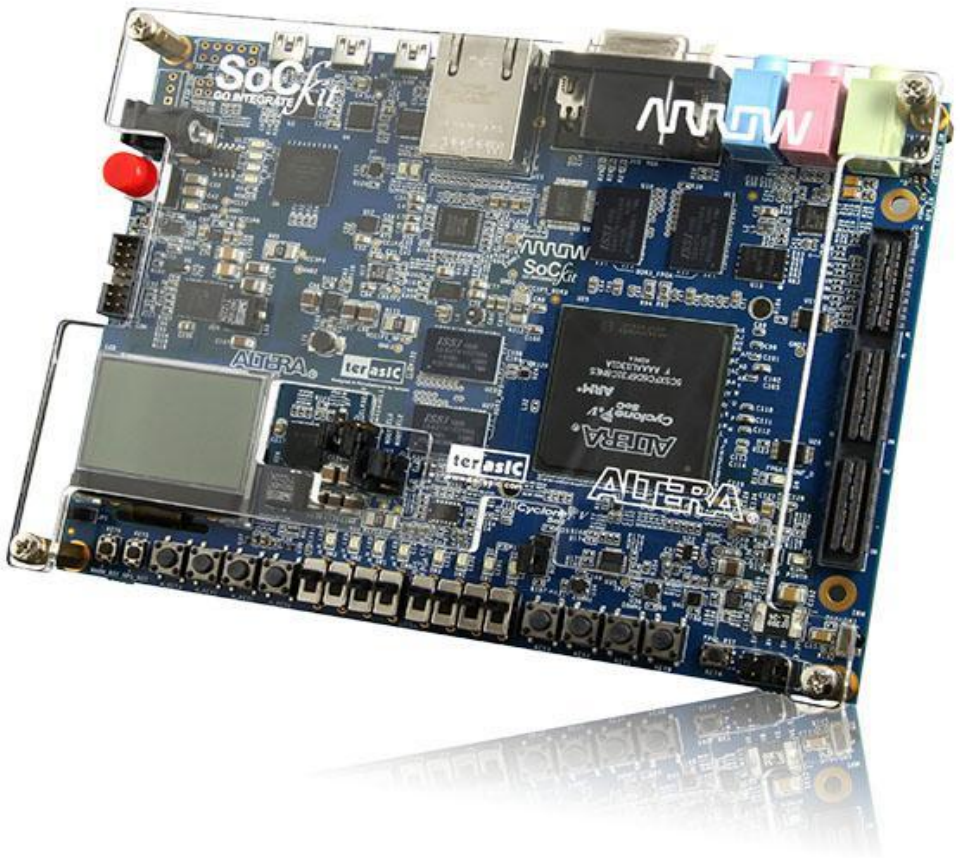
- Малая интегральная схема (МИС) — до 100 элементов
- Средняя интегральная схема (СИС) — до 1000
- Большая интегральная схема (БИС) — до 10000
- ***Сверхбольшая интегральная схема(СБИС)— до 1 млн.***
- Ультрабольшая интегральная схема (УБИС) — до 1 млрд
- Гигабольшая интегральная схема(ГБИС) — более 1 млрд



Определение ПЛИС

Программируемая логическая интегральная схема *(ПЛИС)* - электронный компонент (интегральная микросхема), используемый для создания конфигурируемых цифровых электронных схем.

- *Высокая надежность*
- *Низкое энергопотребление*
- *Интеграция встроенного CPU*
- *Жесткое реальное время*
- *Параллелизм вычислений и работы блоков*



Языки описания аппаратуры Verilog и VHDL

Verilog , *VHDL* - языки описания моделей аппаратуры, пригодных для дальнейшего автоматического синтеза спецификаций, с целью производства реальных чипов.

Verilog

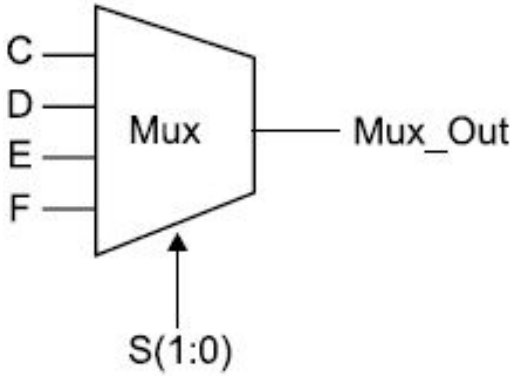
- *Простота конструкций;*
- *параллелизм;*
- *Схожесть с Си;*
- *Слабо типизирован*
- *Нет разделения интерфейса и реализации*

VHDL

- *Более сложные конструкции;*
- *параллелизм;*
- *большая универсальность (описание цифровых схем и других моделей) ;*
- *строго типизирован*
- *Разделение интерфейса и реализации*

Языки описания аппаратуры Verilog и VHDL

Пример реализации простого мультиплексора с применением Verilog и VHDL



VHDL

```
entity mux is
        port (c, d, e, f:       in std_logic;
              s:                in std_logic_vector(1 downto 0);
              mux_out:          out std_logic);
end mux;

architecture mux_impl of mux is
begin
        muxl: process (s, c, d, e, f)
        begin
            case s is
                when “00” => mux_out <= c;
                when “01” => mux_out <= d;
                when “10” => mux_out <= e;
                when others => mux_out <= f;
            end case;
        end process muxl;
end mux_impl;
```

Verilog

```
module mux (c, d, e, f, s, mux_out);
     input      c, d, e, f;
     input      [1:0] s;
     output     mux_out;
     reg        mux_out;

     always @ (c or d or e or f or s)
     begin
        case (s)
          2’b00: mux_out = c;
          2’b01: mux_out = d;
          2’b10: mux_out = e;
          default: mux_out = f;
        endcase
     end
endmodule
```

Средства разработки и моделирования для ПЛИС

ПО разработки – **Quartus II (Prime)**

Поддерживаемые языки – **Verilog, System Verilog, VHDL, AHDL**

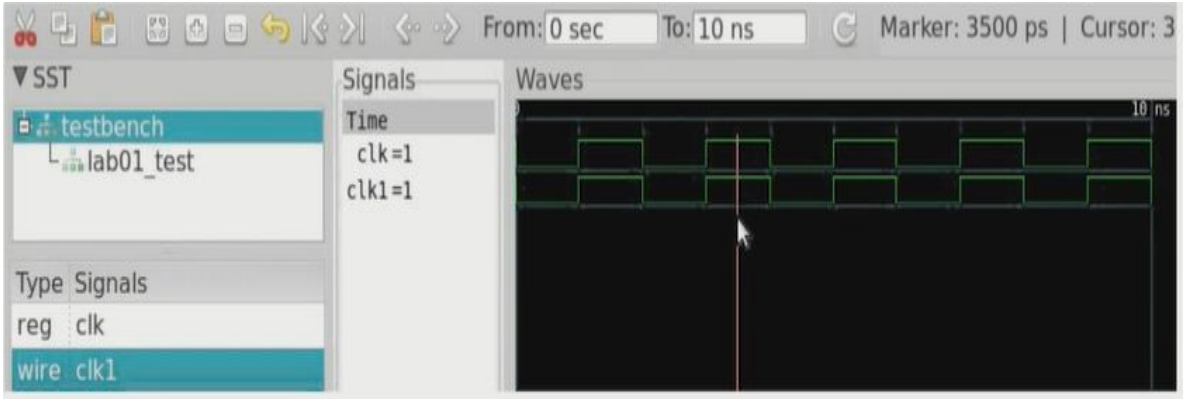
ПО разработки – **ISE Suite, Vivaldo Design Suite**

Поддерживаемые языки – **Verilog, System Verilog, VHDL**

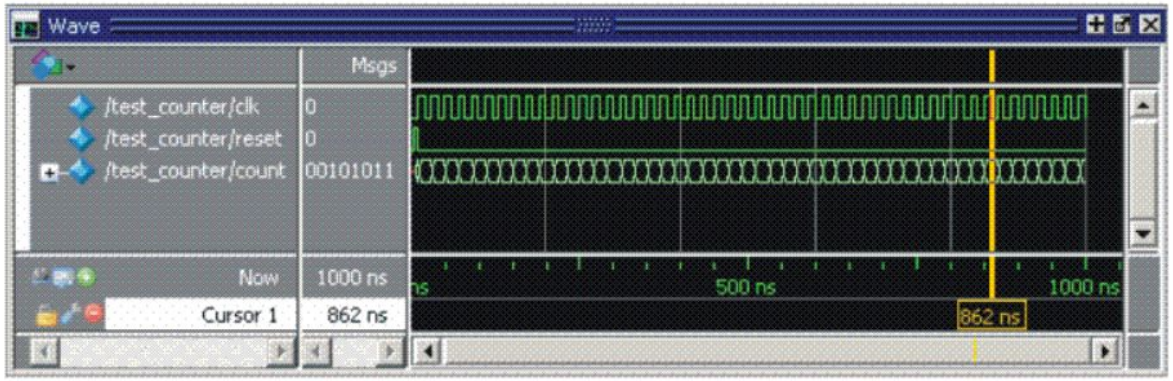
Средства разработки и моделирования для ПЛИС

Программы – симуляторы для ПЛИС

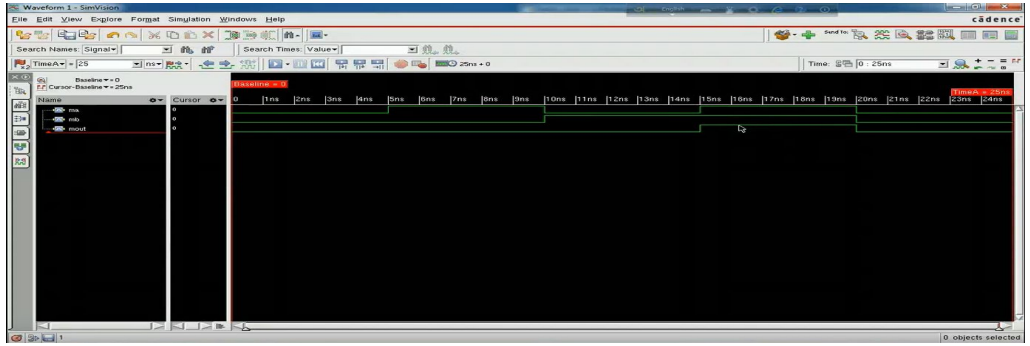
- *Icarus Verilog*



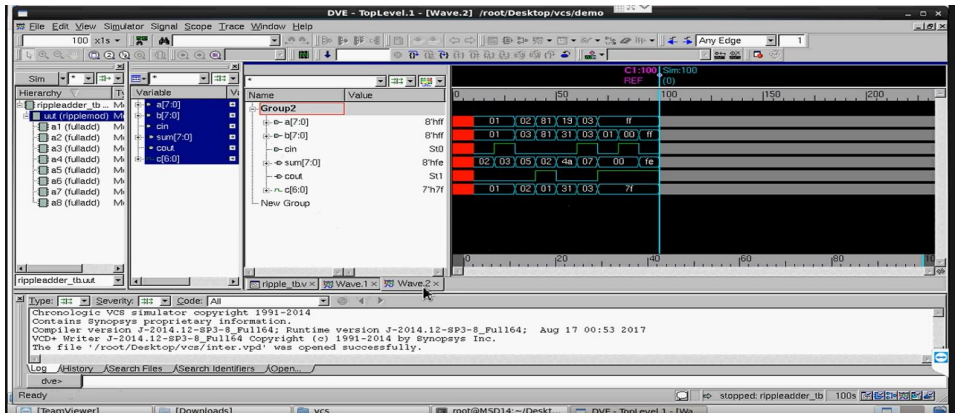
- *Modelsim/QuestaSim – Mentor Graphics*



- *Cadence Ncsim*



- *Synopsys VCS*



Средства разработки и моделирования для ПЛИС

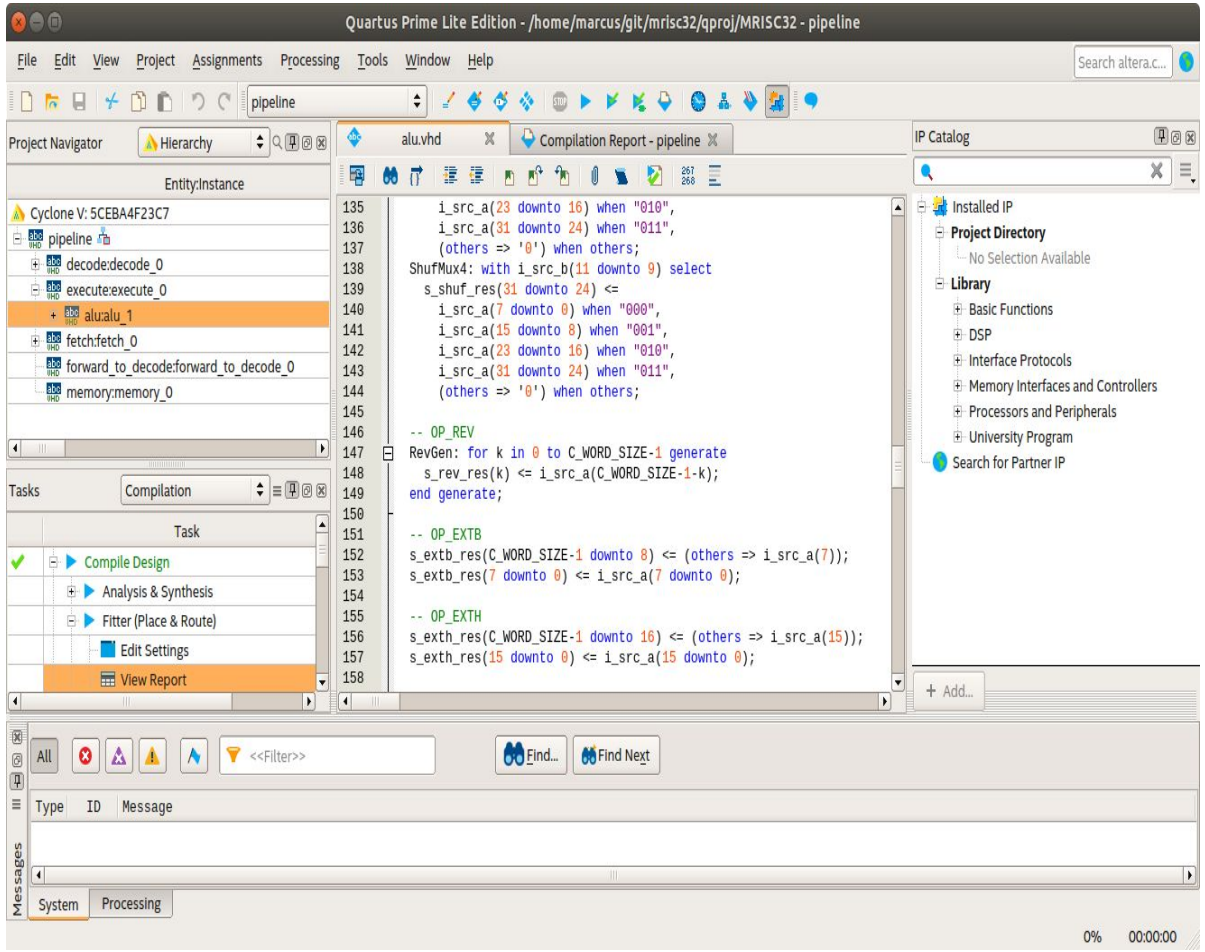
Intel Quartus Prime - для разработки дизайна систем на базе Intel FPGA, SoC и Complex Programmable Logic Device (CPLD), начиная с самых основ и включая далее отладку взаимодействия, оптимизацию, верификацию и моделирование.

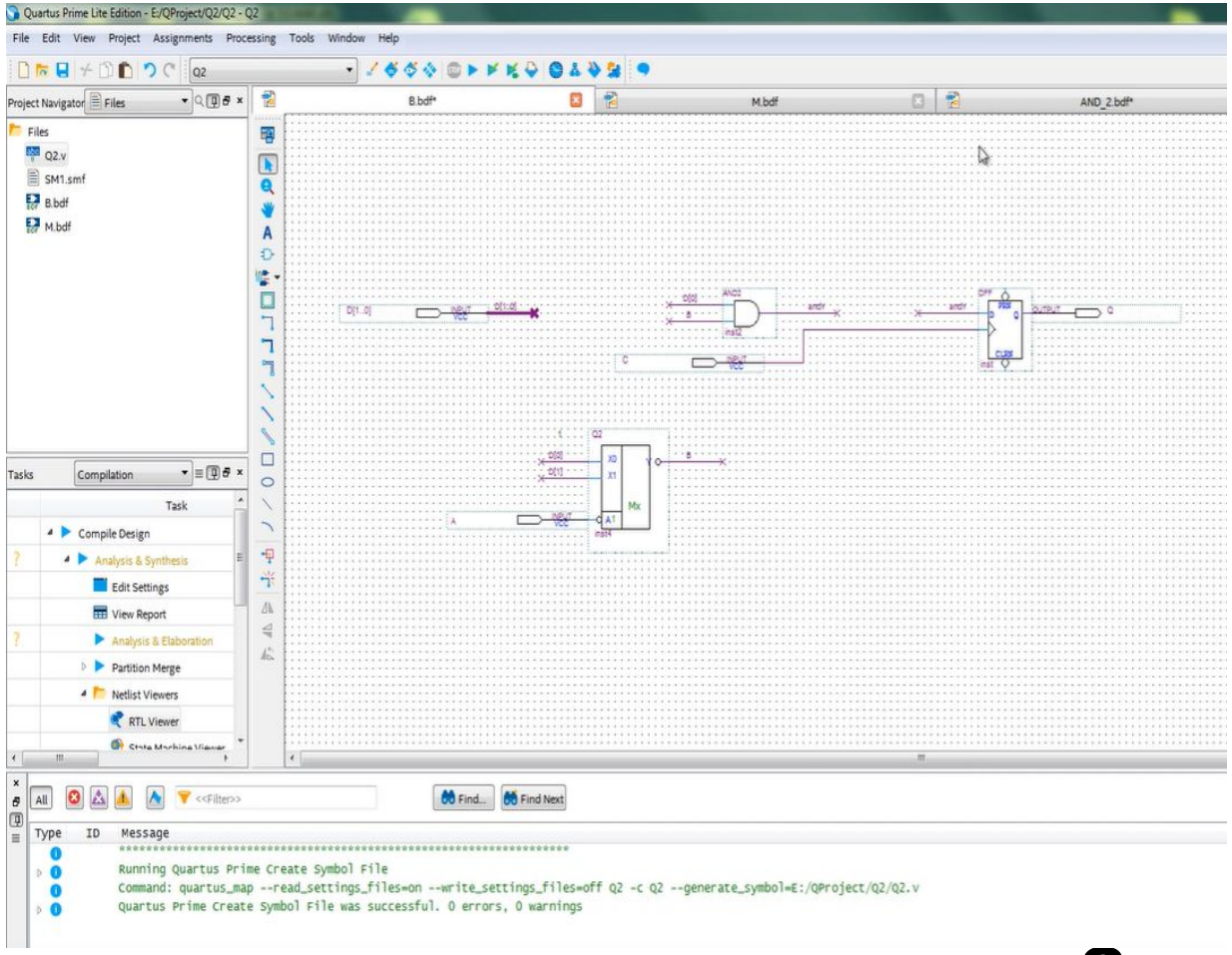
Возможности Quartus

- *Создание схемы графическим способом (Shematic)*
- *Описание на HDL – языках (Verilog, VHDL)*
- *Моделирование работы синтезируемой схемы (ModelSim)*
- *Компиляция проекта, взаимодействие с реальным железом*

Средства разработки и моделирования для ПЛИС

ПО для работы с ПЛИС Altera - Quartus





Пример моделирования работы проекта на Verilog

Quartus Prime Standard Edition - D:/GitHub/Altera/encoder/top - top

File Edit View Project Assignments Processing Tools Window Help

top

Project Navigator | Hierarchy

Entity:Instance

MAX II: EPM240T100C3

top

top.sv | top_tb.sv

```
1   module top_tb( );
2       timeunit 1ns;
3       timeprecision 1ns;
4
5       logic clock, a_reset, encoder_a, encoder_b;
6       logic [ 9 : 0 ] coordinate;
7
8       top DUT( clock, a_reset, encoder_a, encoder_b, coordinate);
9
10      initial begin
11          a_reset = 0;
12          clock = 1;
13          encoder_a = 0;
14          encoder_b = 0;
15          #10ns;
16          a_reset = 1;
17          #1us;
18          $stop;
19      end
20
21      always begin
22          encoder_a = 1; encoder_b = 0; #20ns;
23          encoder_b = 1; #20ns;
24          encoder_a = 0; #20ns;
25          encoder_b = 0; #20ns;
26      end
27
28      always begin
29          #5ns clock = ~clock;
30      end
31
32  endmodule
33
```

Tasks | Compilation

Task

Compile Design

Пример моделирования работы проекта на Verilog

Quartus Prime Standard Edition - D:/GitHub/Altera/encoder/top - top

File Edit View Project Assignments Processing Tools Window Help

top.sv | top_tb.sv

```
1   module top( input logic clock, a_reset, encoder_a, encoder_b, output logic [ 9 : 0 ] coordinate );
2
3       typedef enum logic [ 1 : 0 ] { FORWARD, BACK, IDLE, ERROR } direction;
4       direction dir;
5       typedef logic [ 1 : 0 ] state;
6       state current_inputs, previous_inputs;
7
8       always_ff @( posedge clock or negedge a_reset ) begin
9           if ( ~a_reset ) begin
10              coordinate = '0;
11          end else begin
12              current_inputs <= { encoder_a, encoder_b };
13              previous_inputs <= current_inputs;
14              case ( dir )
15                  IDLE: begin coordinate <= coordinate; end
16                  FORWARD: begin coordinate <= coordinate + 10'b1; end
17                  BACK: begin coordinate <= coordinate - 10'b1; end
18                  ERROR: begin coordinate <= '0; end
19              endcase
20          end
21      end
22
23      always_comb begin
24          case( previous_inputs )
25              2'b00: begin
26                  case ( current_inputs )
27                      2'b00: begin dir = IDLE; end
28                      2'b01: begin dir = BACK; end
29                      2'b10: begin dir = FORWARD; end
30                      2'b11: begin dir = ERROR; end
31                  endcase
32              end
33              2'b01: begin
34                  case ( current_inputs )
35                      2'b00: begin dir = FORWARD; end
36                      2'b01: begin dir = IDLE; end
37                      2'b10: begin dir = ERROR; end
38                      2'b11: begin dir = FORWARD; end
39                  endcase
40              end
41              2'b10: begin
42                  case ( current_inputs )
43                      2'b00: begin dir = BACK; end
44                      2'b01: begin dir = ERROR; end
45                      2'b10: begin dir = IDLE; end
46                      2'b11: begin dir = FORWARD; end
47                  endcase
48              end
49              2'b11: begin
50                  case ( current_inputs )
51                      2'b00: begin dir = ERROR; end
52                      2'b01: begin dir = FORWARD; end
53                      2'b10: begin dir = BACK; end
54                      2'b11: begin dir = IDLE; end
55                  endcase
56              end
```

Project Navigator | Hierarchy

Entity:Instance

MAX II: EPM240T100C3

top

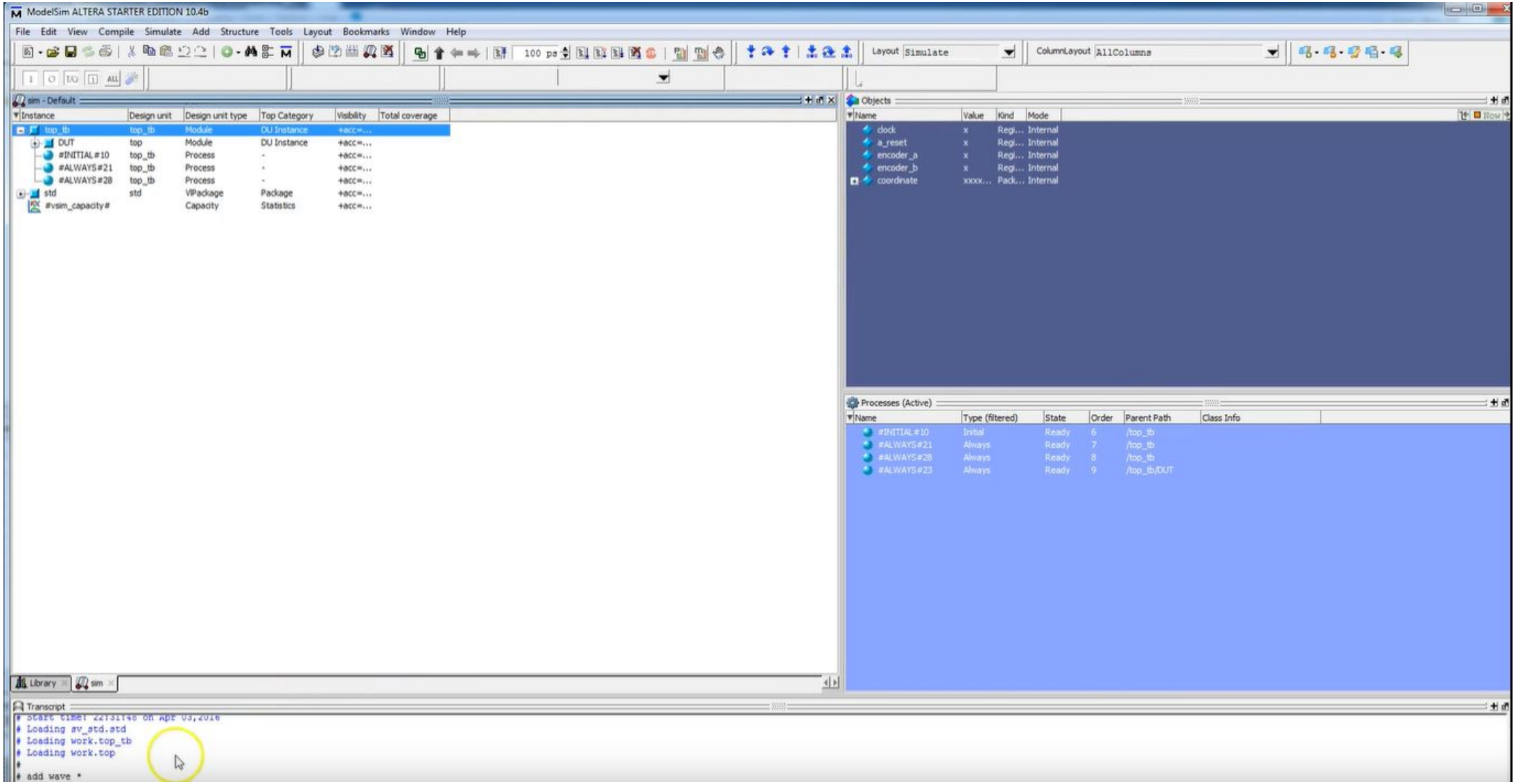
Tasks: Compilation

Task

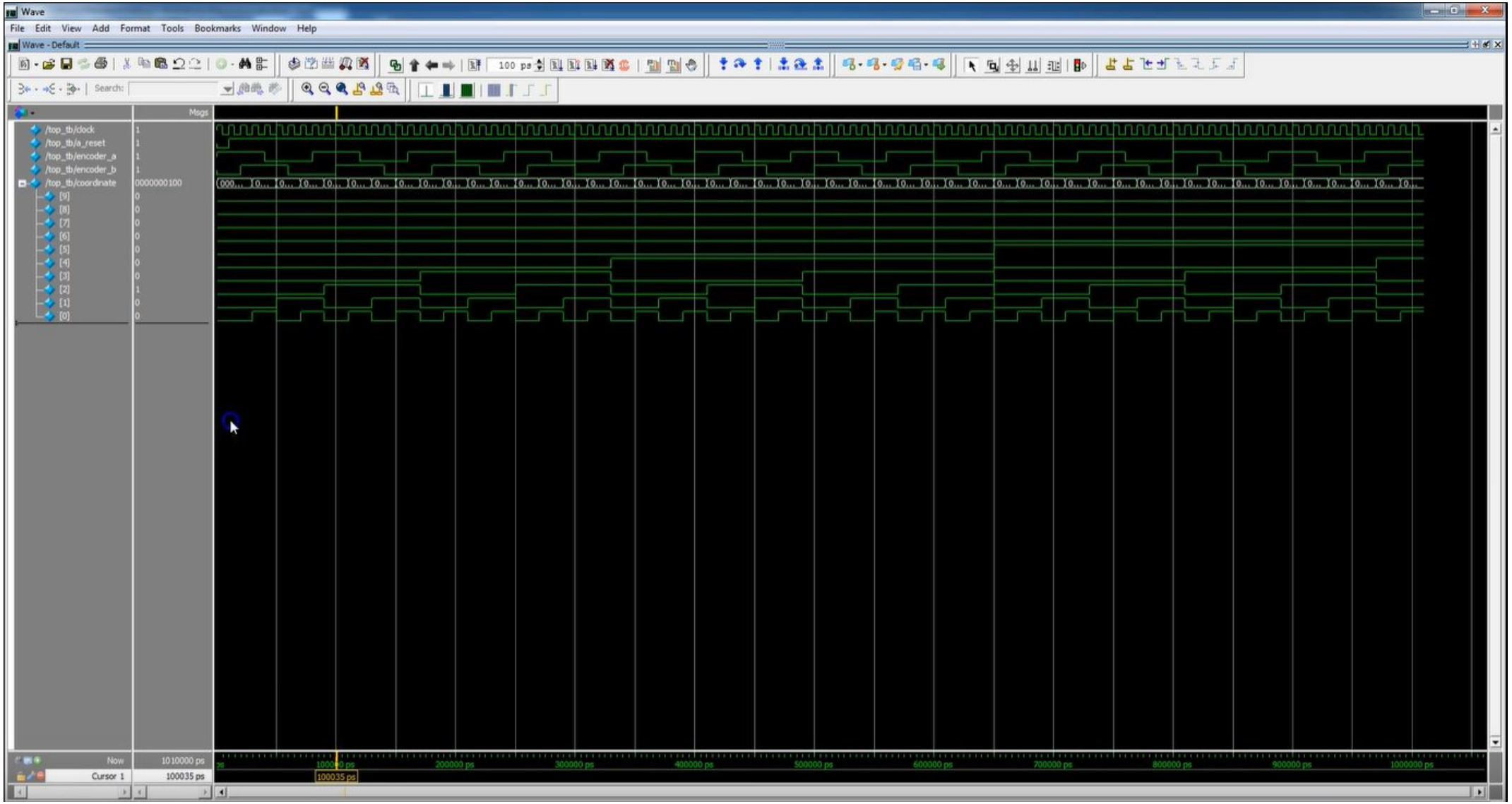
- Compile Design
  - Analysis & Synthesis
  - Fitter (Place & Route)
  - Assembler (Generate programming
  - TimeQuest Timing Analysis
  - EDA Netlist Writer
- Edit Settings
- Program Device (Open Programmer)

Type ID Message

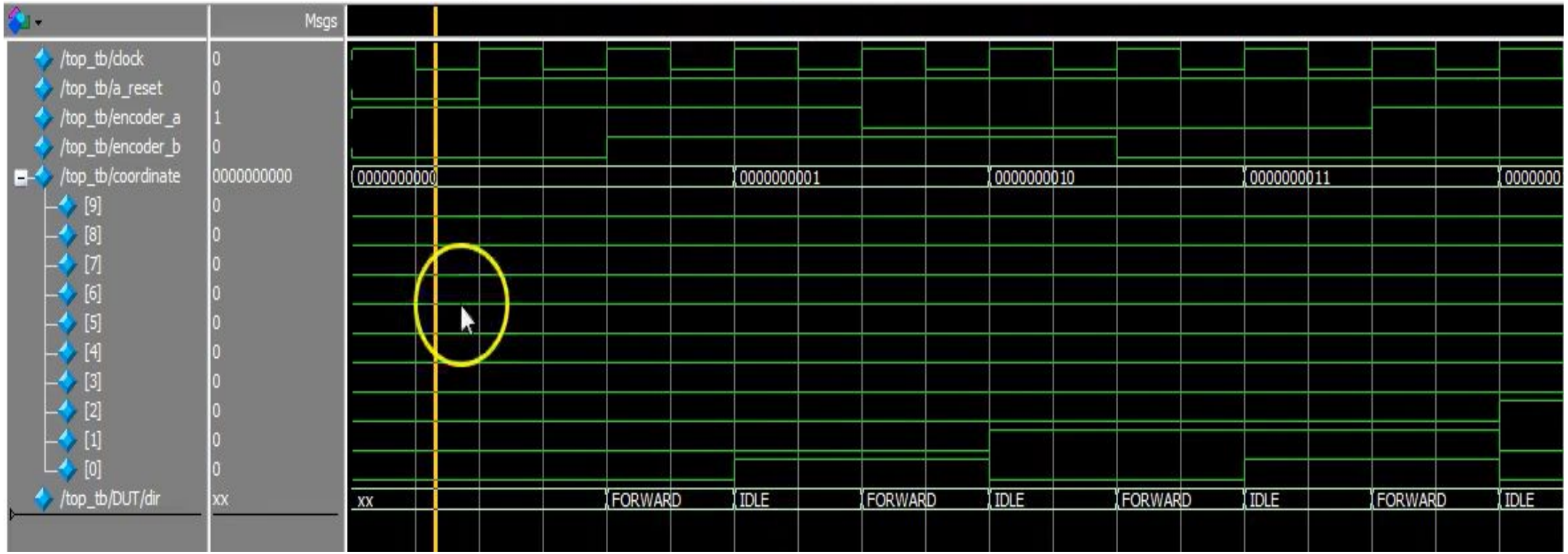
Пример моделирования работы проекта на Verilog



Пример моделирования работы проекта на Verilog



Пример моделирования работы проекта на Verilog



Спасибо за внимание!

Выполнил: Студент гр. 368-М1

Лункин Дмитрий Сергеевич