Кафедра промышленной электроники (ПрЭ)

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

Выполнил: Студент гр. 368-М1

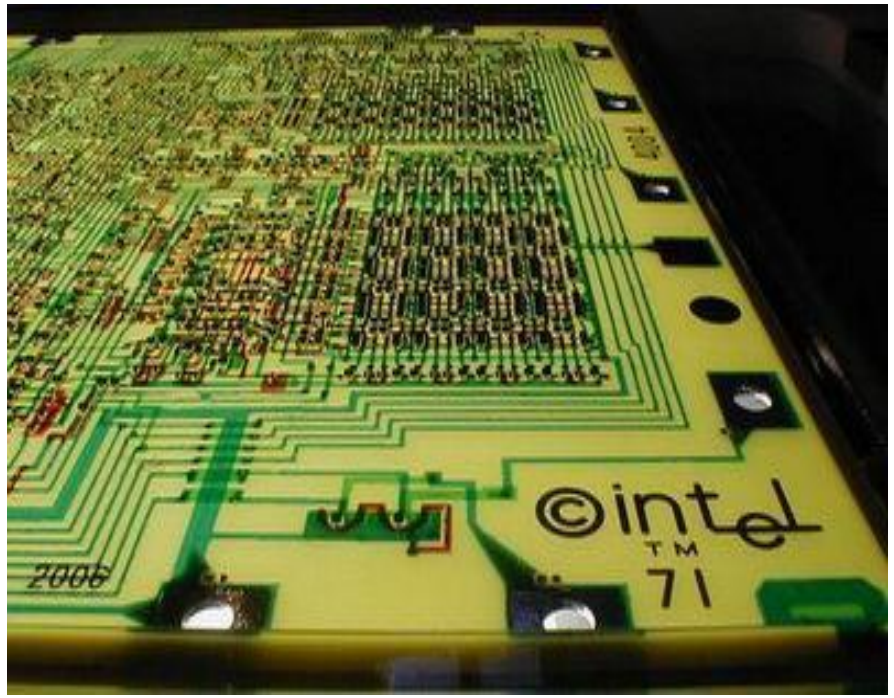
Лункин Дмитрий Сергеевич

Томск 2019

Определение СБИС

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

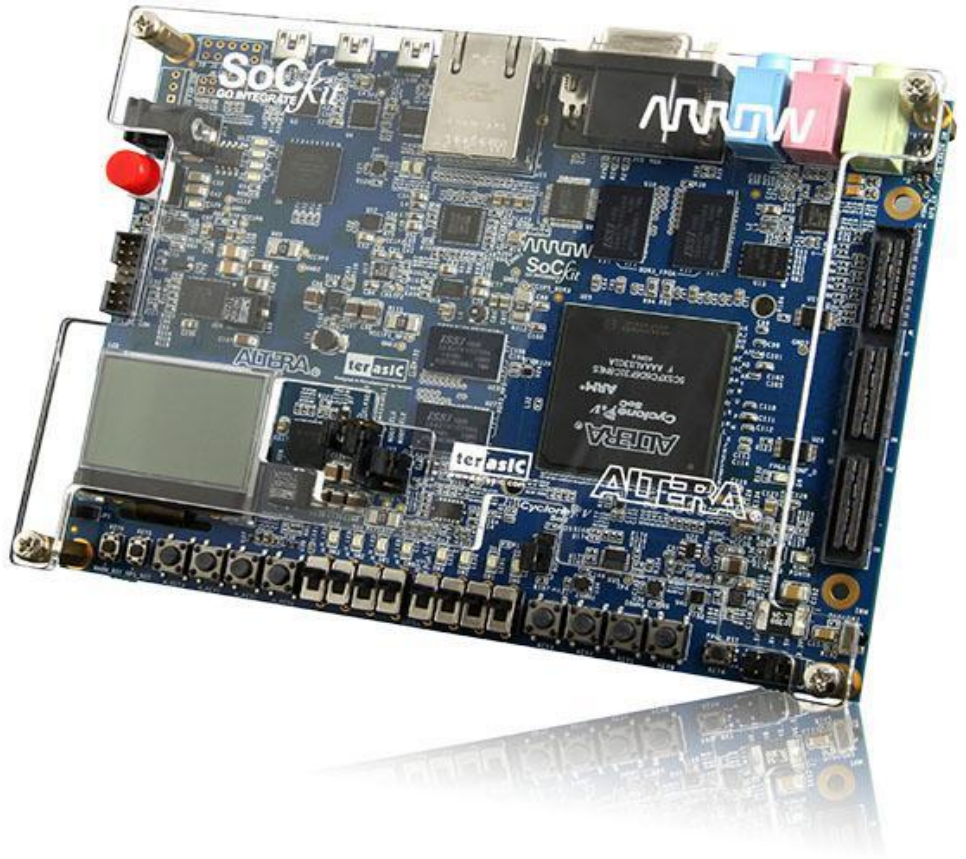
- Малая интегральная схема (МИС) — до 100 элементов
- Средняя интегральная схема (СИС) — до 1000
- Большая интегральная схема (БИС) — до 10000
- ***Сверхбольшая интегральная схема(СБИС)— до 1 млн.***
- Ультрабольшая интегральная схема (УБИС) — до 1 млрд
- Гигабольшая интегральная схема(ГБИС) — более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

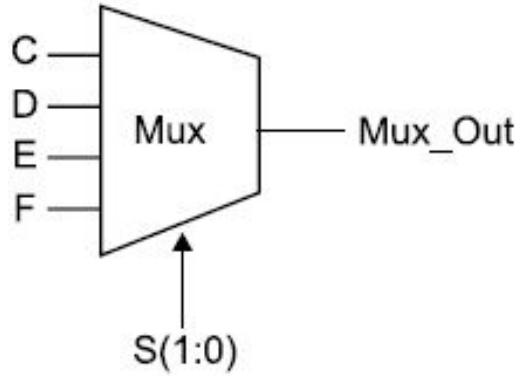
- *Простота конструкций;*
- *параллелизм;*
- *Схожесть с Си;*
- *Слабо типизирован*
- *Нет разделения интерфейса и реализации*

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**

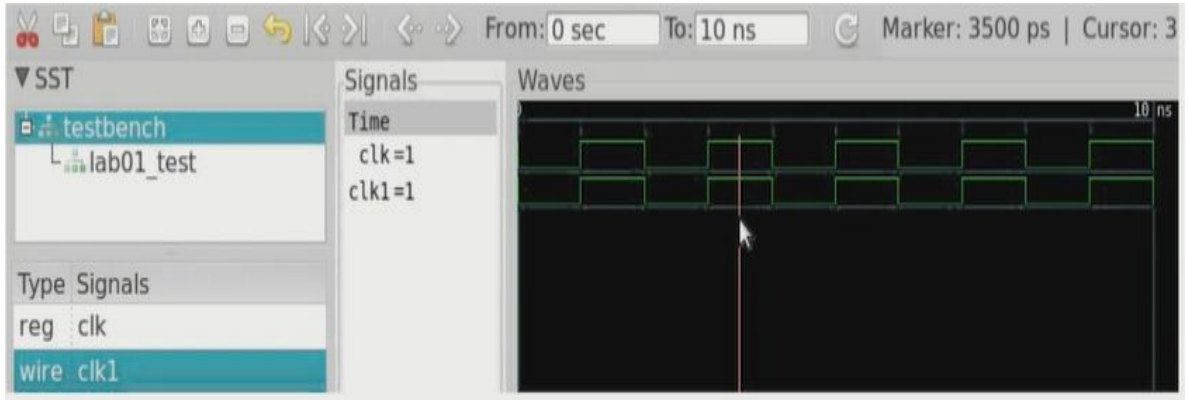
ПО разработки – **ISE Suite, Vivaldo Design Suite**

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

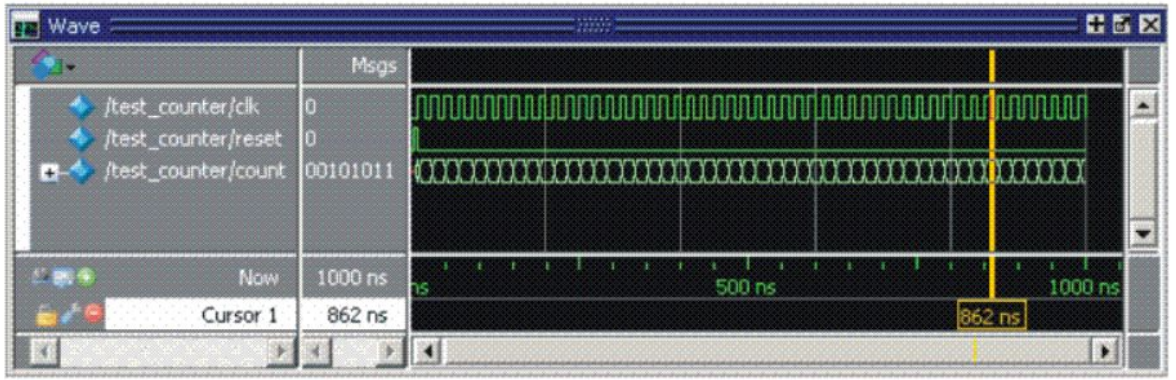
Средства разработки и моделирования для ПЛИС

Программы – симуляторы для ПЛИС

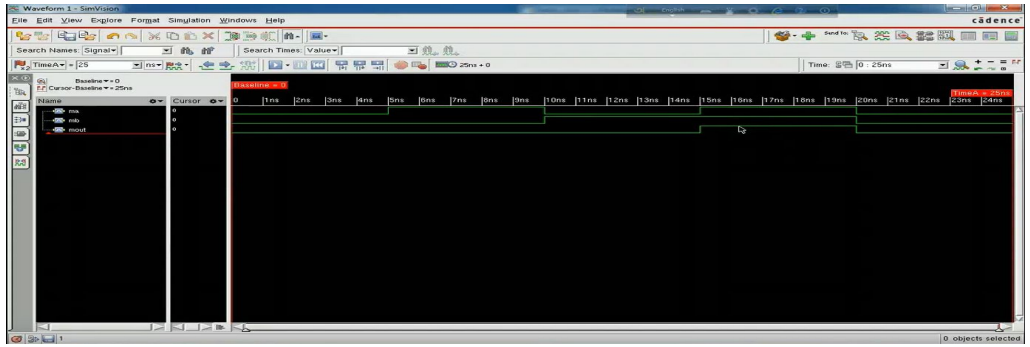
- *Icarus Verilog*



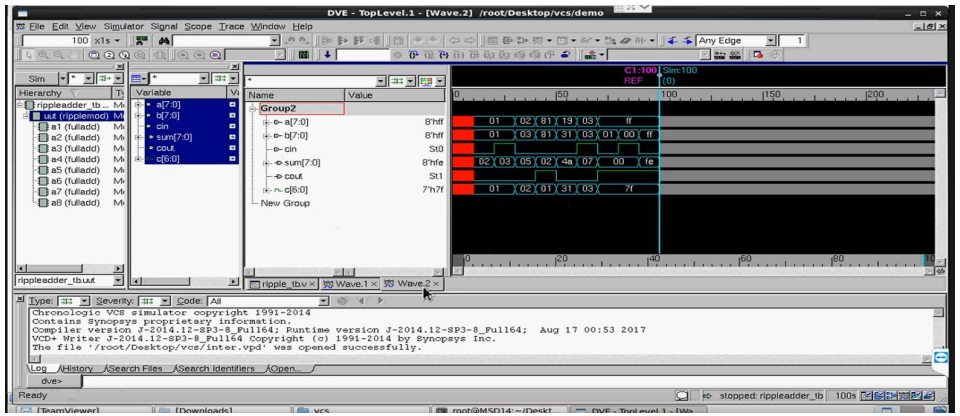
- *Modelsim/QuestaSim – Mentor Graphics*



- *Cadence Ncsim*



- *Synopsys VCS*



Средства разработки и моделирования для ПЛИС

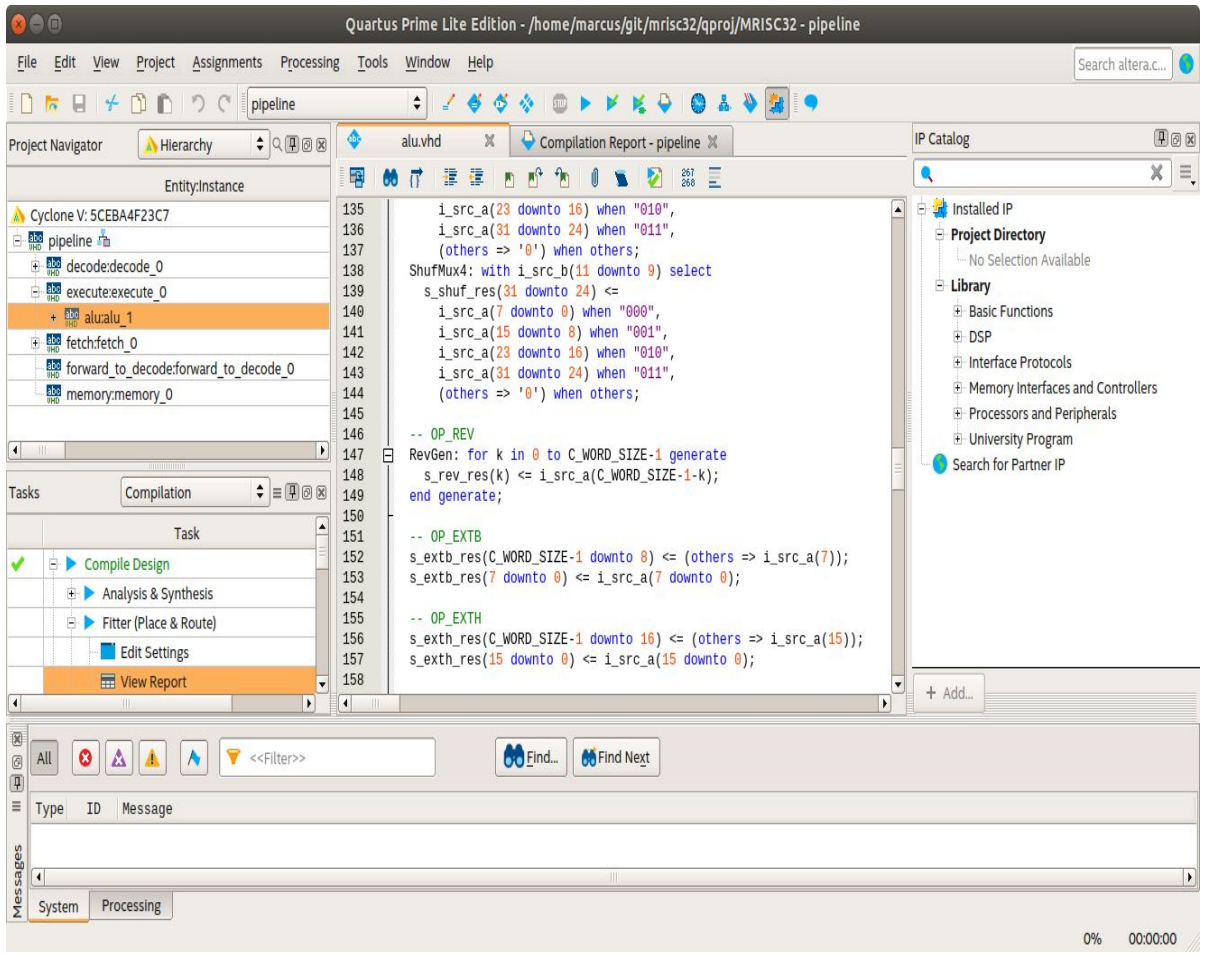
Intel Quartus Prime - для разработки дизайна систем на базе Intel FPGA, SoC и Complex Programmable Logic Device (CPLD), начиная с самых основ и включая далее отладку взаимодействия, оптимизацию, верификацию и моделирование.

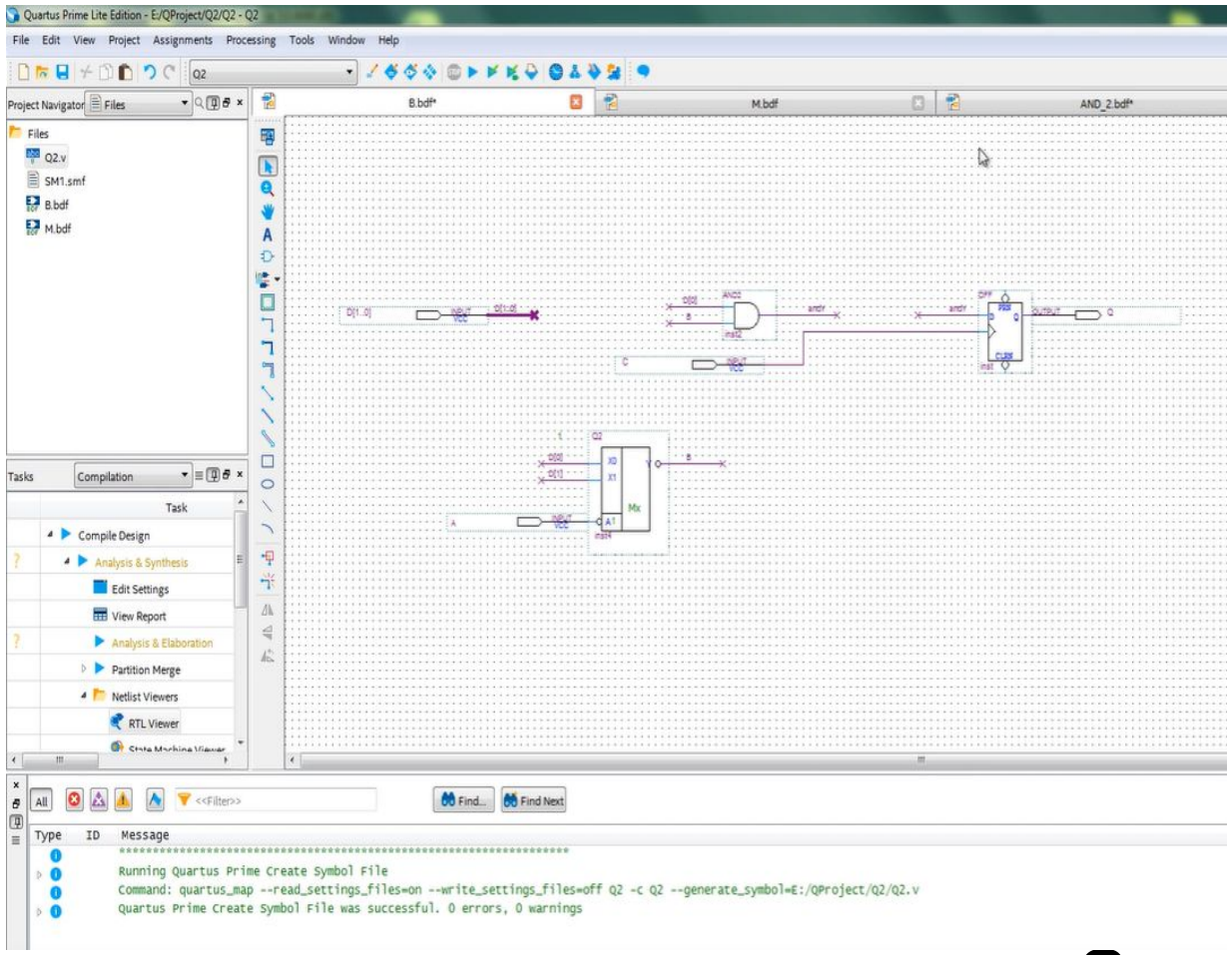
Возможности Quartus

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

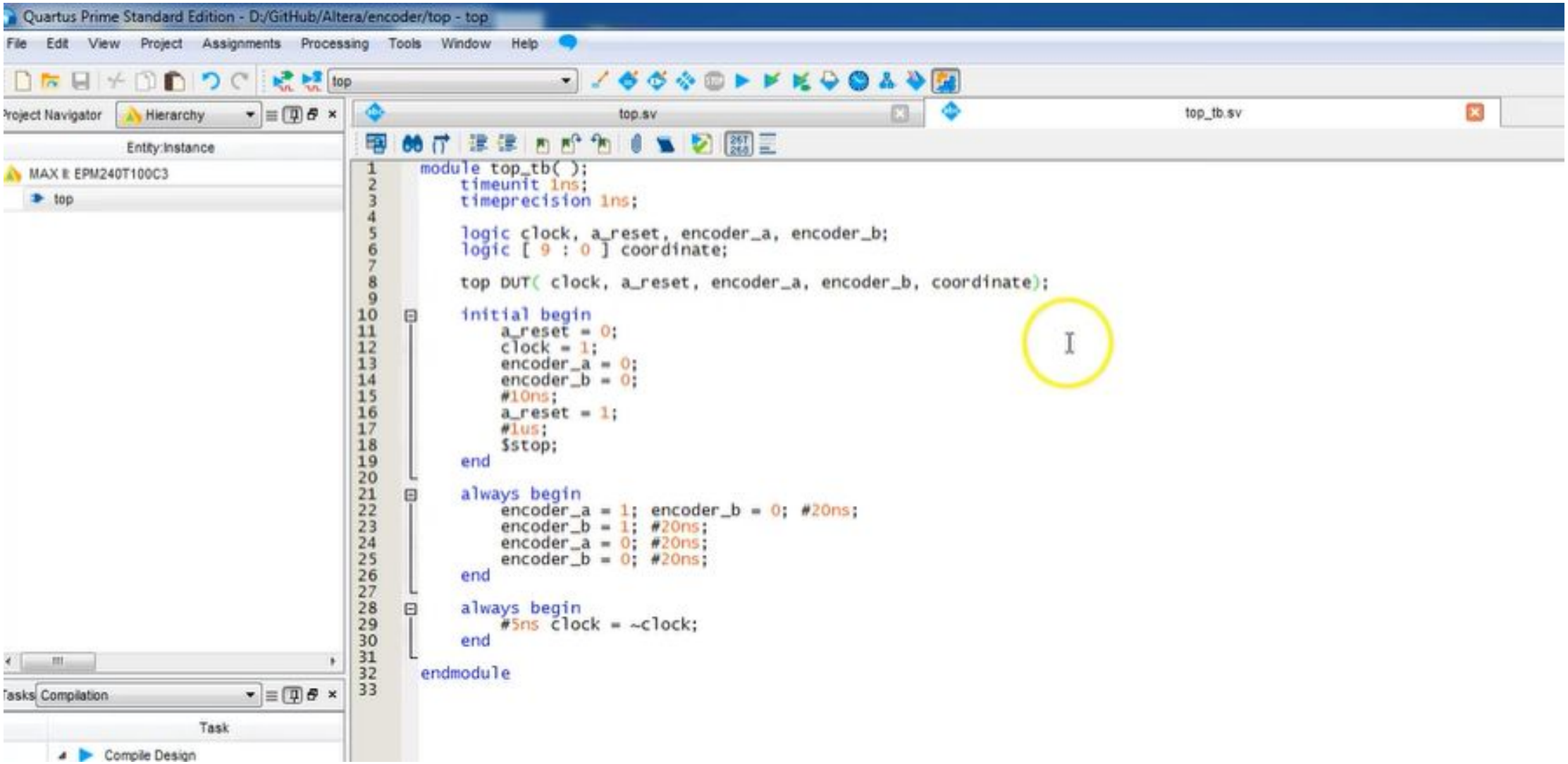
Средства разработки и моделирования для ПЛИС

ПО для работы с ПЛИС Altera - Quartus

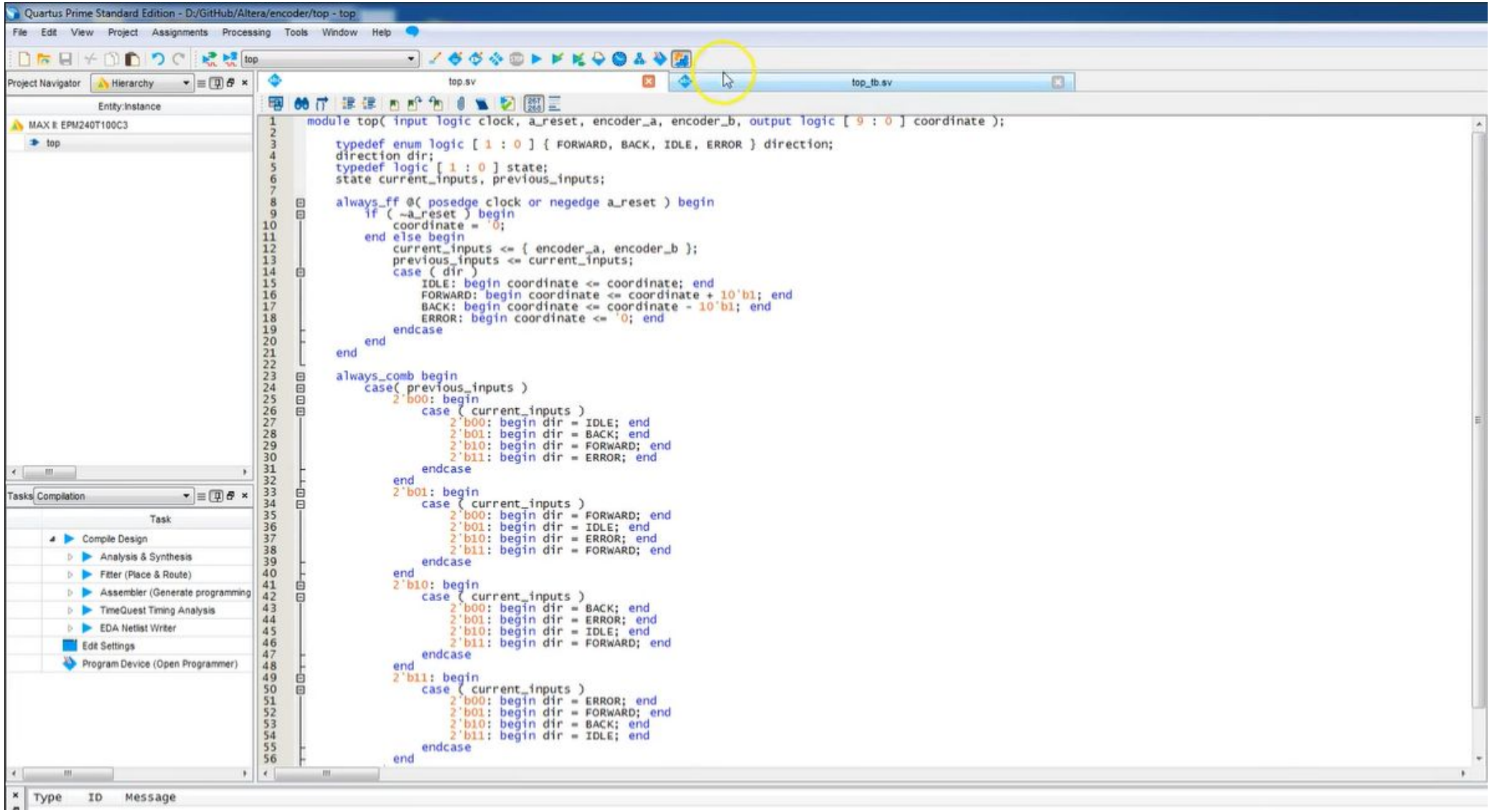




Пример моделирования работы проекта на Verilog

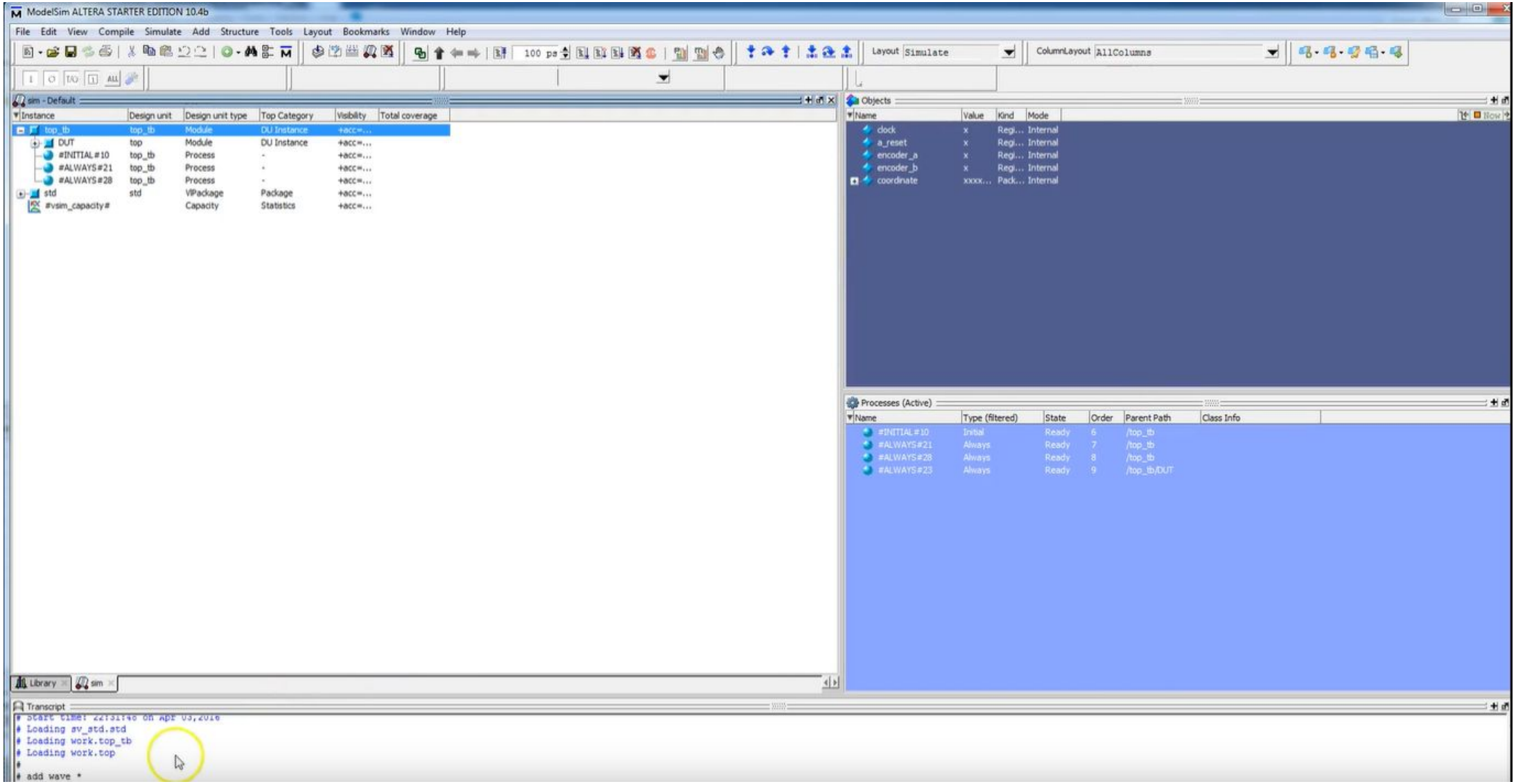


Пример моделирования работы проекта на Verilog

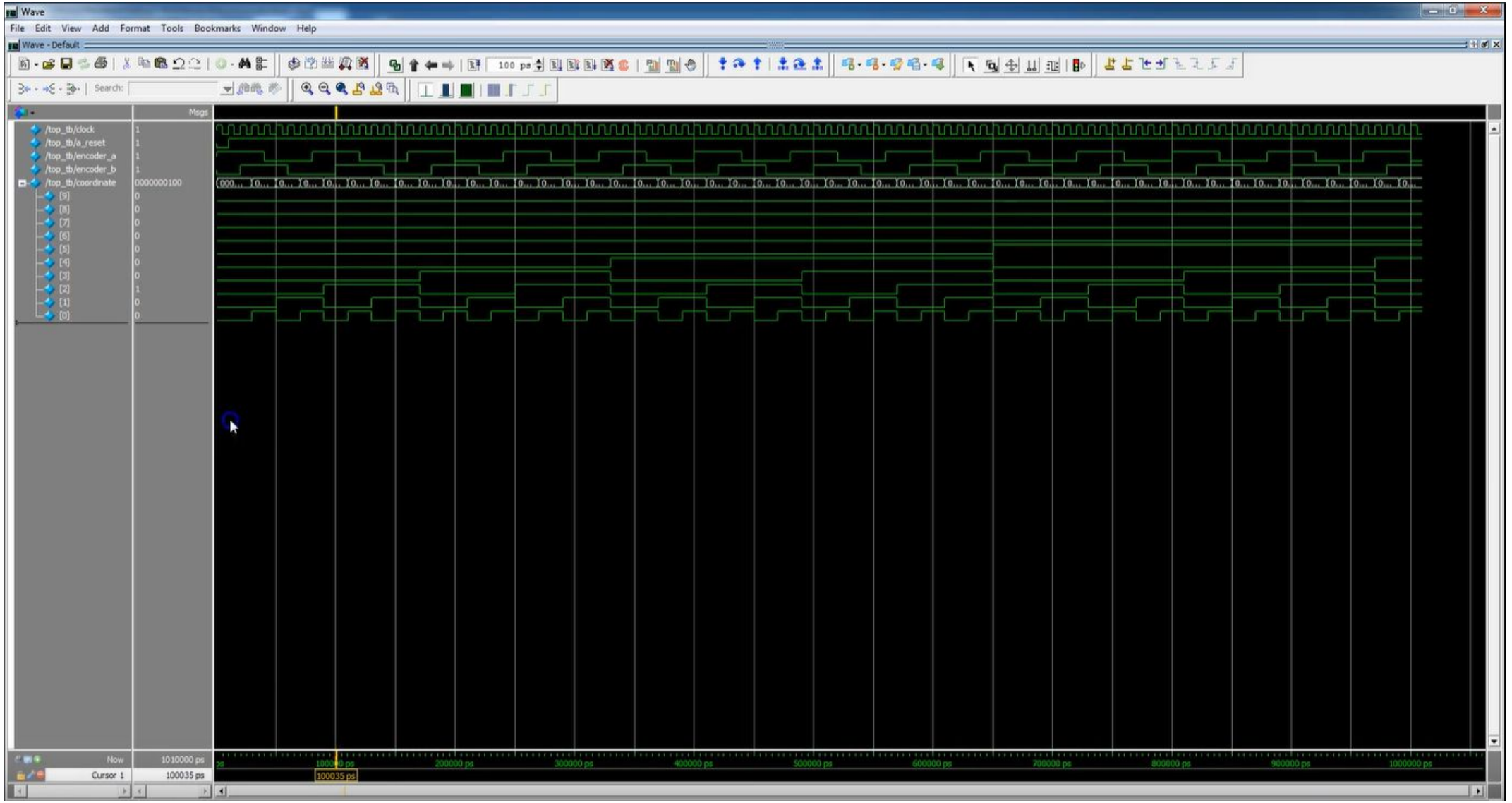



```
module top( input logic clock, a_reset, encoder_a, encoder_b, output logic [ 9 : 0 ] coordinate );

    typedef enum logic [ 1 : 0 ] { FORWARD, BACK, IDLE, ERROR } direction;
    direction dir;
    typedef logic [ 1 : 0 ] state;
    state current_inputs, previous_inputs;

    always_ff @( posedge clock or negedge a_reset ) begin
        if ( ~a_reset ) begin
            coordinate = '0;
        end else begin
            current_inputs <= { encoder_a, encoder_b };
            previous_inputs <= current_inputs;
            case ( dir )
                IDLE: begin coordinate <= coordinate; end
                FORWARD: begin coordinate <= coordinate + 10'b1; end
                BACK: begin coordinate <= coordinate - 10'b1; end
                ERROR: begin coordinate <= '0; end
            endcase
        end
    end

    always_comb begin
        case( previous_inputs )
            2'b00: begin
                case ( current_inputs )
                    2'b00: begin dir = IDLE; end
                    2'b01: begin dir = BACK; end
                    2'b10: begin dir = FORWARD; end
                    2'b11: begin dir = ERROR; end
                endcase
            end
            2'b01: begin
                case ( current_inputs )
                    2'b00: begin dir = FORWARD; end
                    2'b01: begin dir = IDLE; end
                    2'b10: begin dir = ERROR; end
                    2'b11: begin dir = FORWARD; end
                endcase
            end
            2'b10: begin
                case ( current_inputs )
                    2'b00: begin dir = BACK; end
                    2'b01: begin dir = ERROR; end
                    2'b10: begin dir = IDLE; end
                    2'b11: begin dir = FORWARD; end
                endcase
            end
            2'b11: begin
                case ( current_inputs )
                    2'b00: begin dir = ERROR; end
                    2'b01: begin dir = FORWARD; end
                    2'b10: begin dir = BACK; end
                    2'b11: begin dir = IDLE; end
                endcase
            end
```

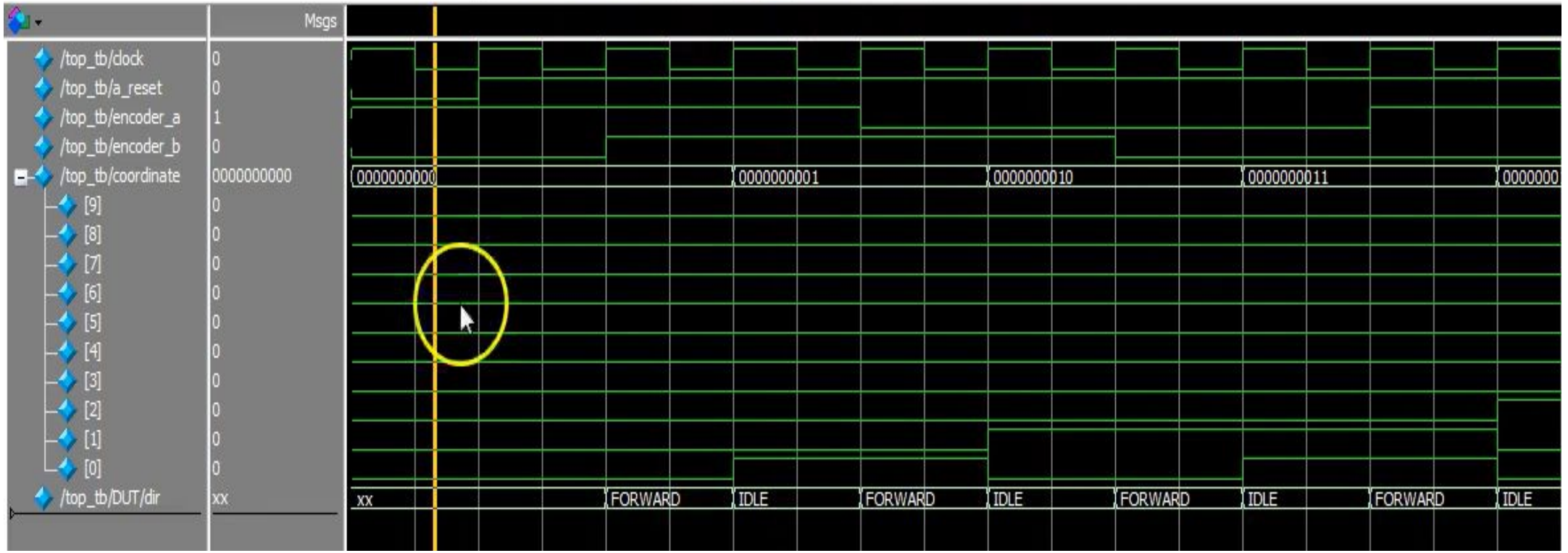
Пример моделирования работы проекта на Verilog



Пример моделирования работы проекта на Verilog



Пример моделирования работы проекта на Verilog



Спасибо за внимание!

Выполнил: Студент гр. 368-М1

Лункин Дмитрий Сергеевич