



Лингвистические средства проектирования

The screenshot displays a VHDL editor interface. At the top, there are tabs for 'counter.vhd', 'example_vhd1.vhd', 'example_vhd2.vhd', and 'example_vhd3.vhd'. The main window shows a timing diagram for a clock signal 'kate' and a VHDL code snippet. Below the code, a logic circuit diagram is shown, illustrating the implementation of a counter using logic gates (AND, OR, XOR) and flip-flops. The circuit includes inputs A and B, and outputs A+B and A⊕B. The VHDL code defines an entity 'counter' with a generic parameter 'n' and ports for 'clock', 'clear', 'count', and 'Q'.

```
entity counter is
  generic (n : natural := 2);
  port (
    clock : in std_logic;
    clear : in std_logic;
    count : in std_logic;
    Q : out std_logic_vector(n-1 downto 0);
  );
end counter;
```

Лекция 7

Моделирование и синтез

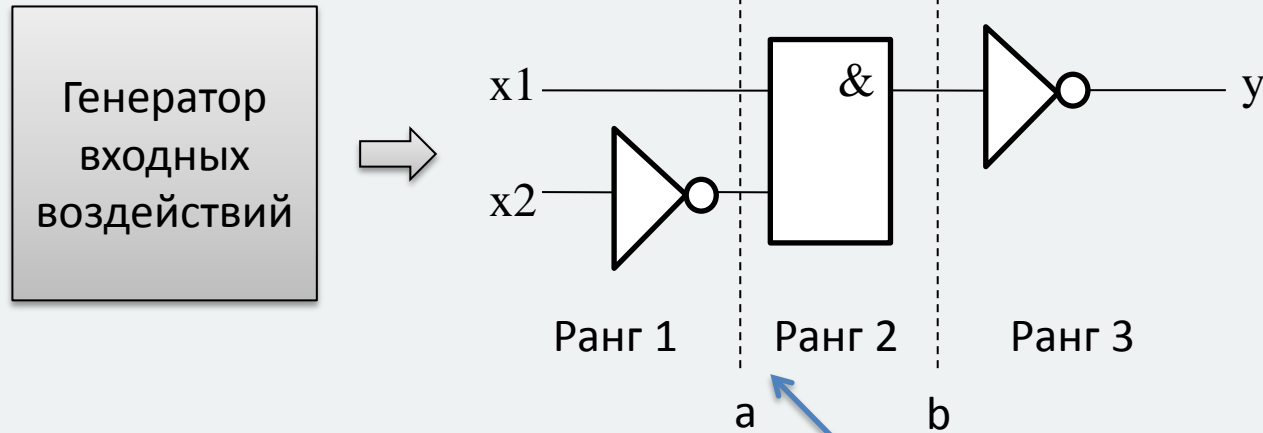
Развитие HDL языков



Классификация видов моделирования



Сквозное моделирование



1. $a = \text{not } x2$
2. $b = x1 \text{ and } a$
3. $y = \text{not } b$

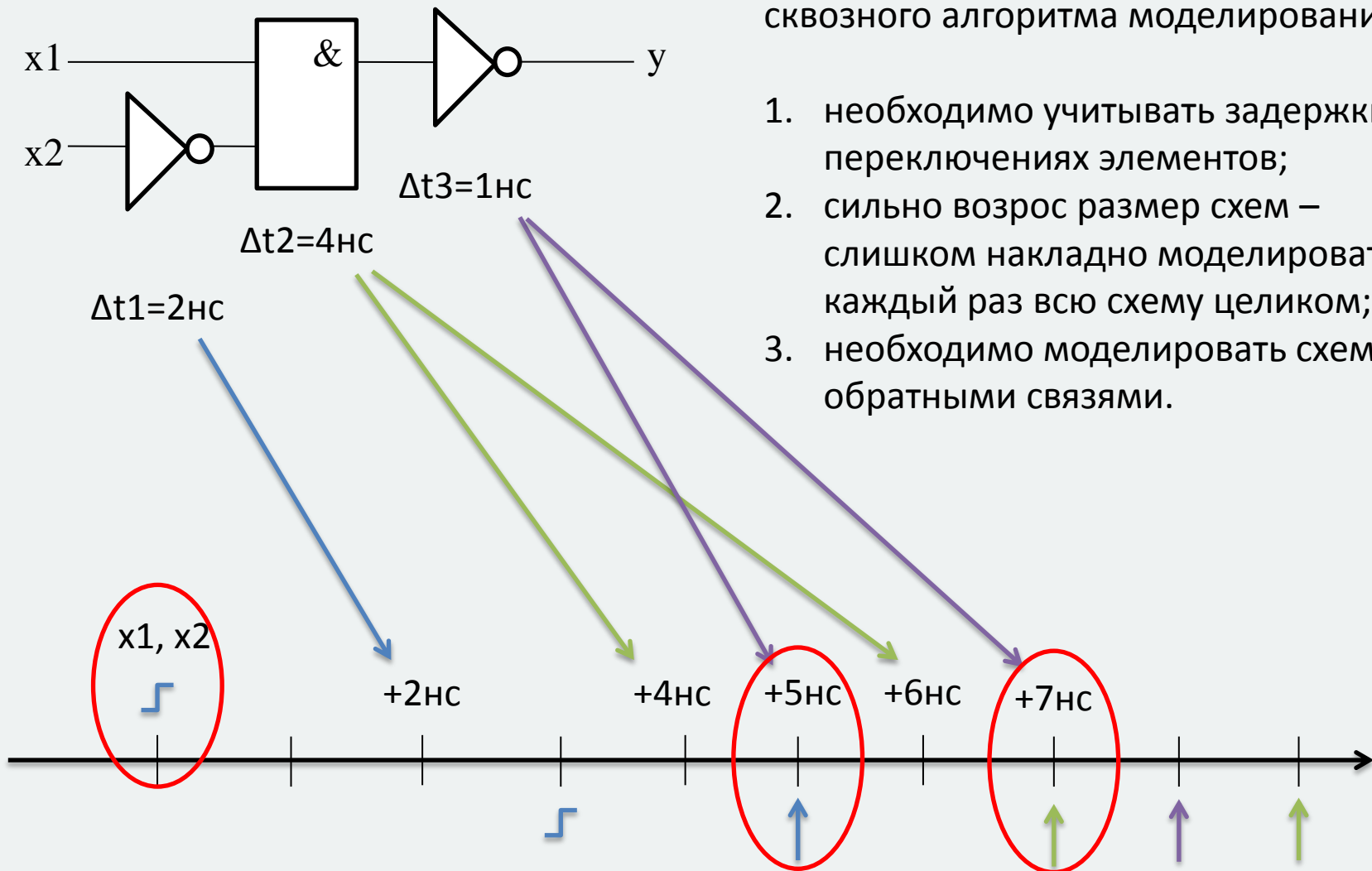
Базовый сквозной алгоритм

- Провести ранжирование схемы
- Сформировать ММ
- Провести моделирование

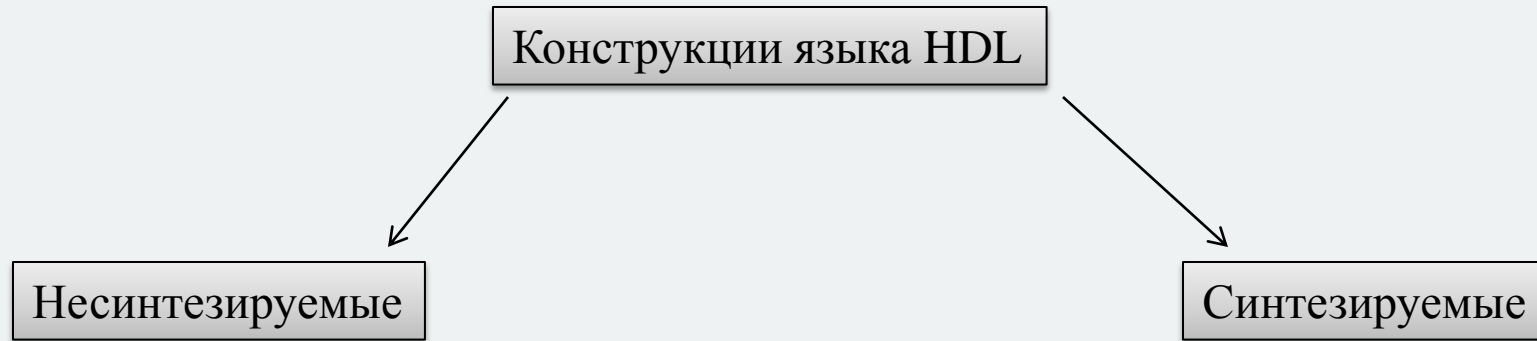
Событийное моделирование (1)

Основные предпосылки реализации сквозного алгоритма моделирования :

1. необходимо учитывать задержки в переключениях элементов;
2. сильно возрос размер схем – слишком накладно моделировать каждый раз всю схему целиком;
3. необходимо моделировать схемы с обратными связями.



Формы HDL кода



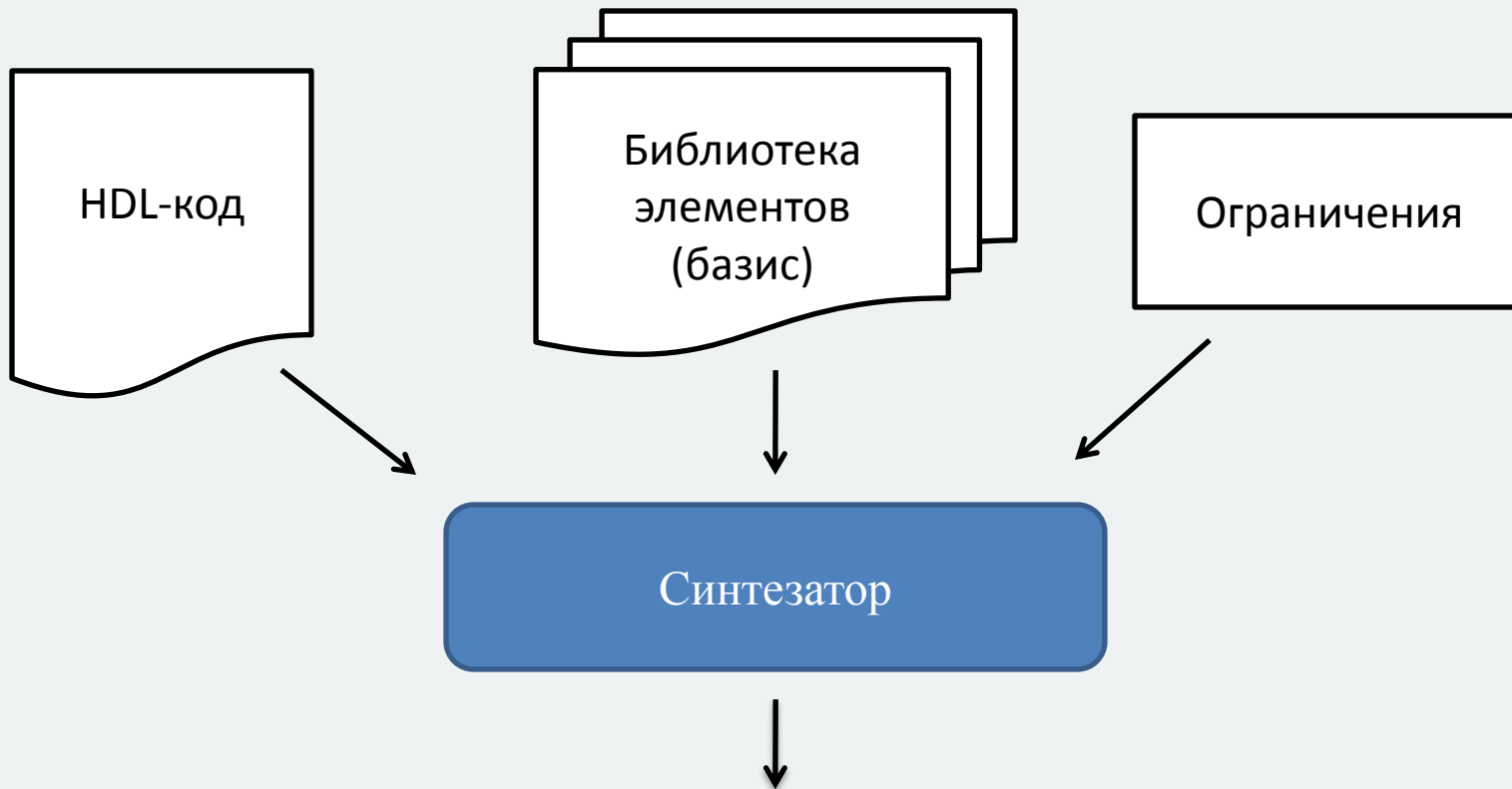
Возможно проведение функциональной и временной верификации.

Для получения топологии нужно:

1. переписывать HDL-код,
2. создавать схемотехническое представление вручную;
3. формировать топологию вручную.

Возможно проведение функциональной и временной верификации, автоматического синтеза схемотехнического представления и топологии .

Работа программ синтеза



1. RTL HDL-описание;
2. нетлист на аналоговом языке – spice, spectre;
3. топологическое представление схемы.

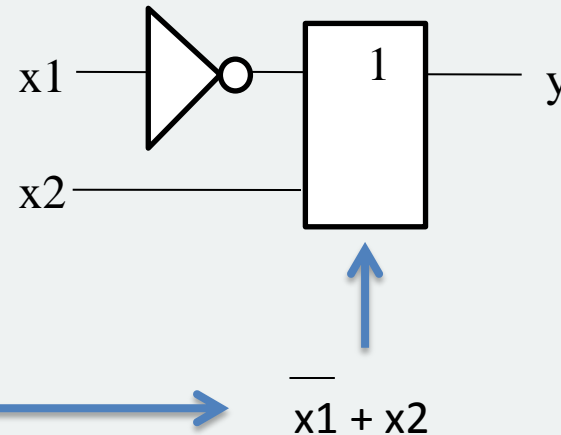
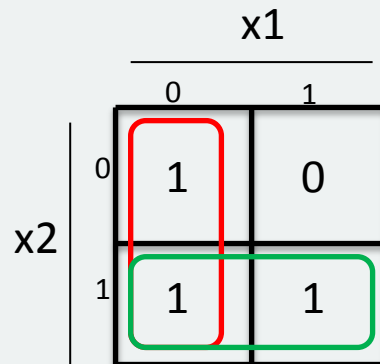
Синтез комбинационных схем методом карт Карно (1)

```
architecture BEH of DEVICE is
begin
  process (x1, x2)
  begin
    if (x1='1' and x2='0') then
      y <= '0';
    else
      y <= '1';
    end if;
  end process;
end BEH;
```



↓

x1	x2	y
0	0	1
0	1	1
1	0	0
1	1	1



Базис – «ИЛИ» - смотрим по единицам

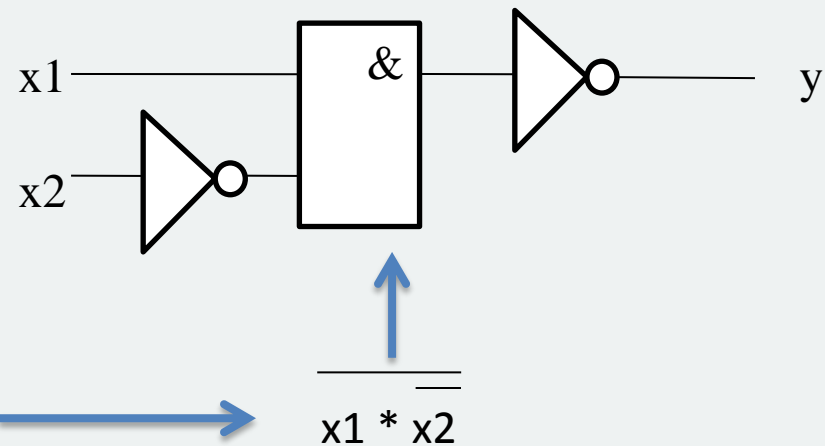
Синтез комбинационных схем методом карт Карно (2)

```
architecture BEH of DEVICE is
begin
  process (x1, x2)
  begin
    if (x1='1' and x2='0') then
      y <= '0';
    else
      y <= '1';
    end if;
  end process;
end BEH;
```



x1	x2	y
0	0	1
0	1	1
1	0	0
1	1	1

	x1	
	0	1
0	1	0
1	1	1



Базис – «И» - смотрим по нулям, выход инвертируем