



Цифровые устройства и микропроцессоры I часть

Лекция 14

Лектор: Богаченков Алексей Николаевич

e-mail: microproc@mail.ru

Темы лекции:

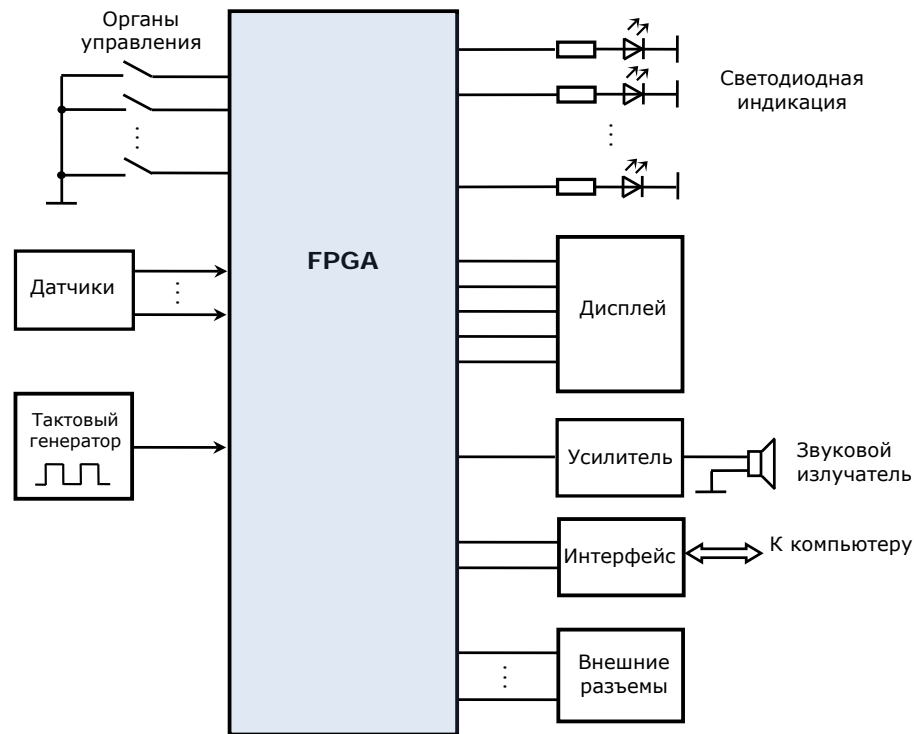
Основные этапы проектирования
устройств на ПЛИС

Основные элементы языка VHDL

Примеры описаний на языке VHDL

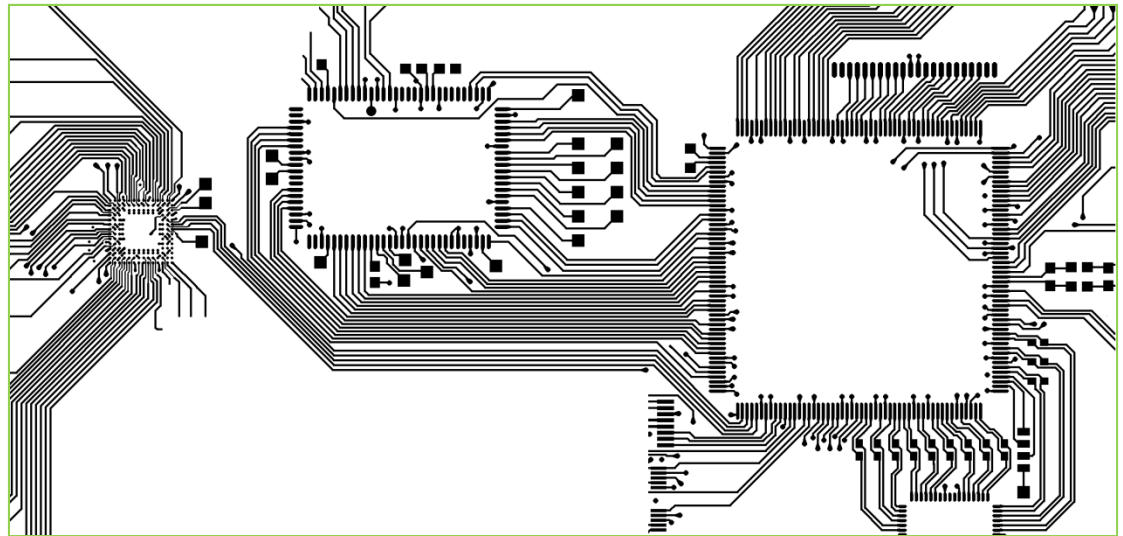
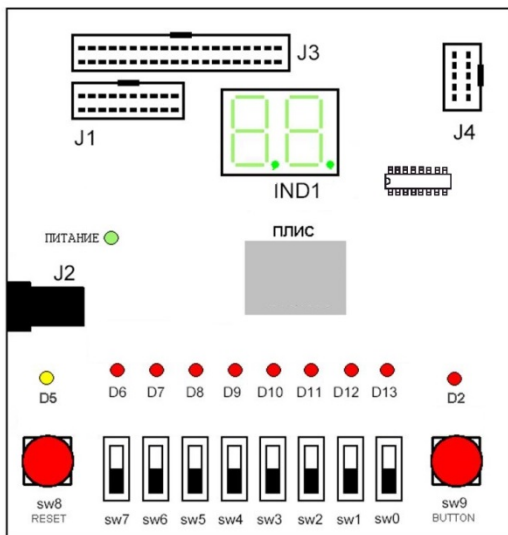
Основные этапы проектирования устройств на ПЛИС

- Постановка задачи (техническое задание).
- Составление схемы (структурной, функциональной, принципиальной), где ПЛИС представлена как микросхема.



Основные этапы проектирования устройств на ПЛИС

- Конструкторский этап. Разработка печатного узла, критерии: удобство подключения, минимум пересечений печатных проводников.

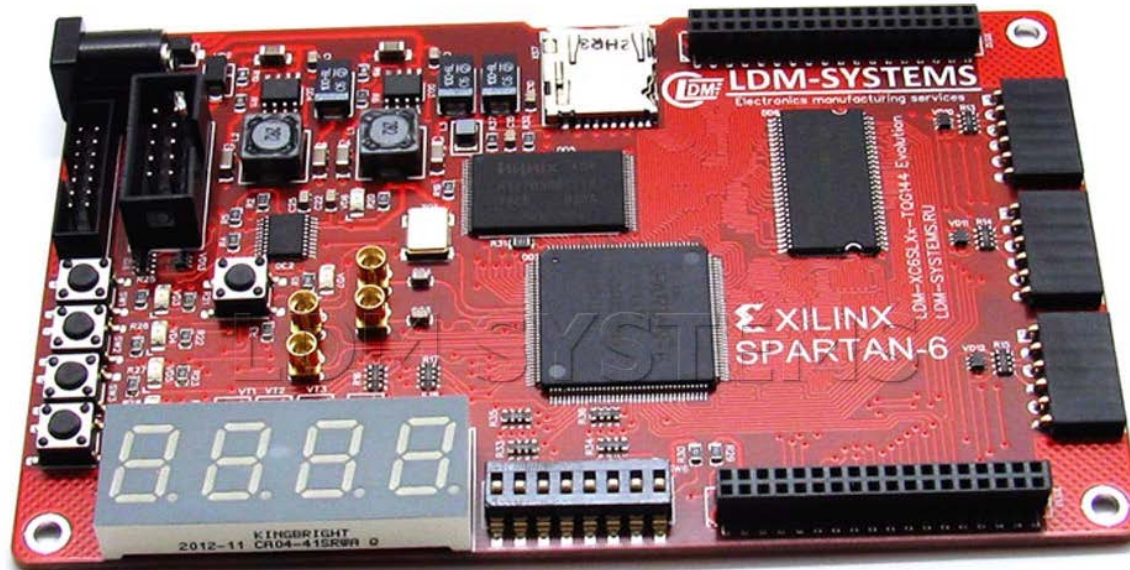
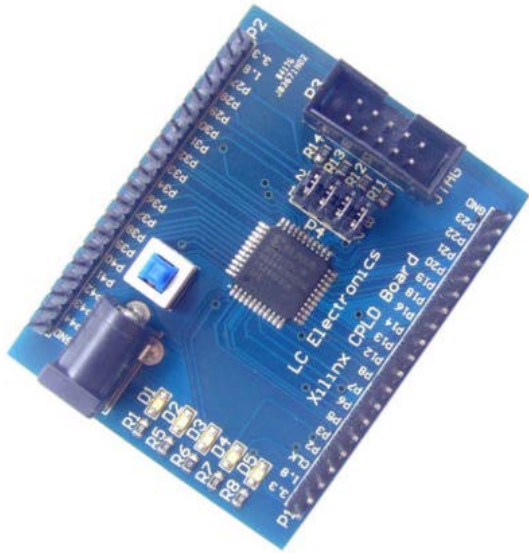


По результатам размещения и трассировки уточняются номера задействованных выводов ПЛИС.

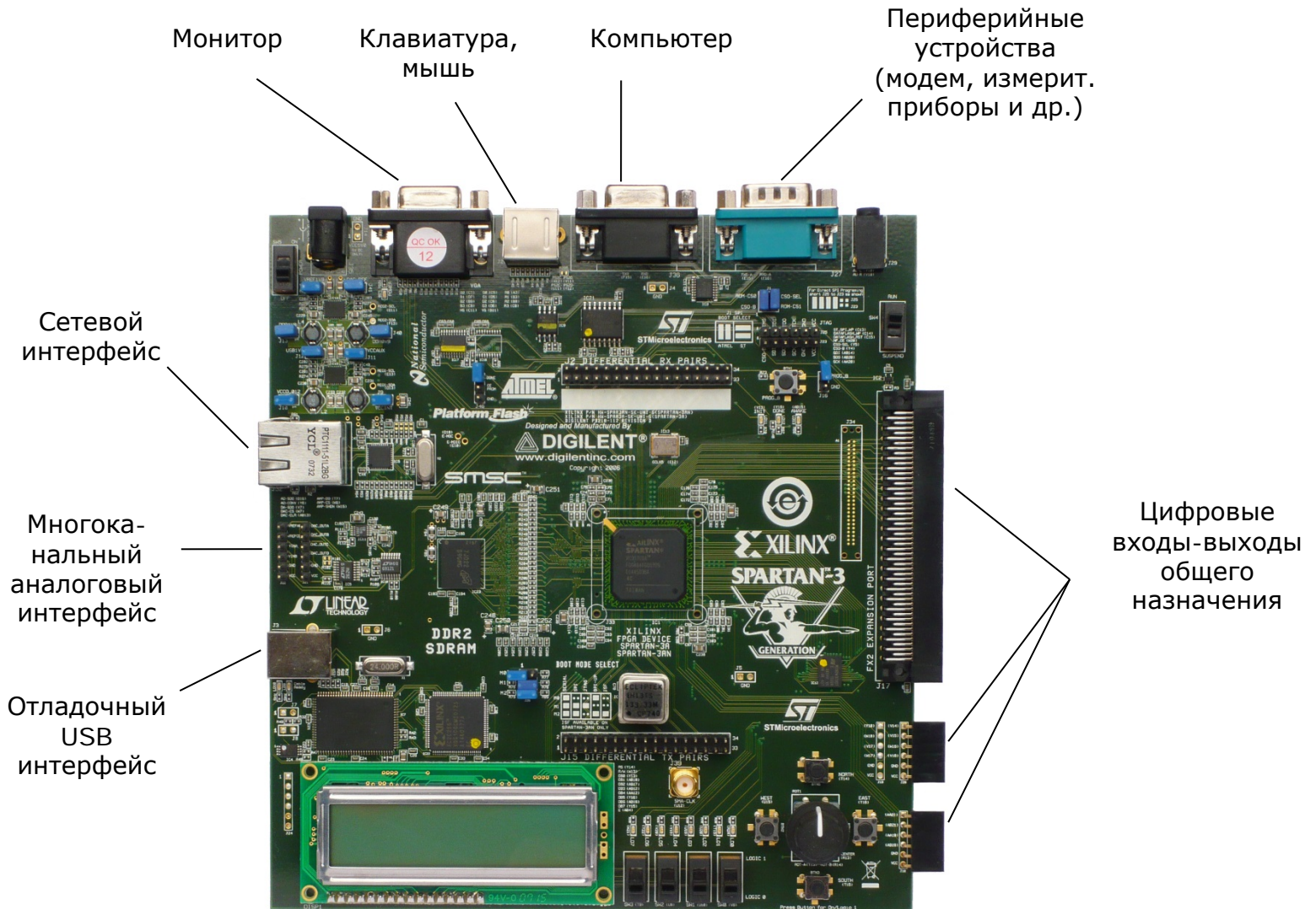
Основные этапы проектирования устройств на ПЛИС

- Создание проекта в САПР ПЛИС (семейство/тип ПЛИС, корпус, средства синтеза, проектные ограничения).
- Описание проектируемого устройства (части, размещаемой в кристалле) либо в **схемотехнической форме**, т.е. в виде принципиальной схемы, либо в форме **поведенческого описания** на языке HDL (Hardware Description Language).
- Функциональное моделирование (симуляция) на уровне операторов языка HDL (при необходимости).
- Синтез устройства на уровне регистров (модулей типа дешифраторов, мультиплексоров, счетчиков и т.п.).
- Синтез модулей и всего устройства на кристалле на основе имеющихся ресурсов конкретной ПЛИС: логических ячеек, устройств ввода-вывода, памяти, интерфейсов, IP ядер и т.п.
- Размещение и трассировка проекта в кристалле.
- Временное моделирование (при необходимости).
- Генерация конфигурационного файла, программирование ПЛИС (загрузка проекта в кристалл).

Отладочные платы на ПЛИС

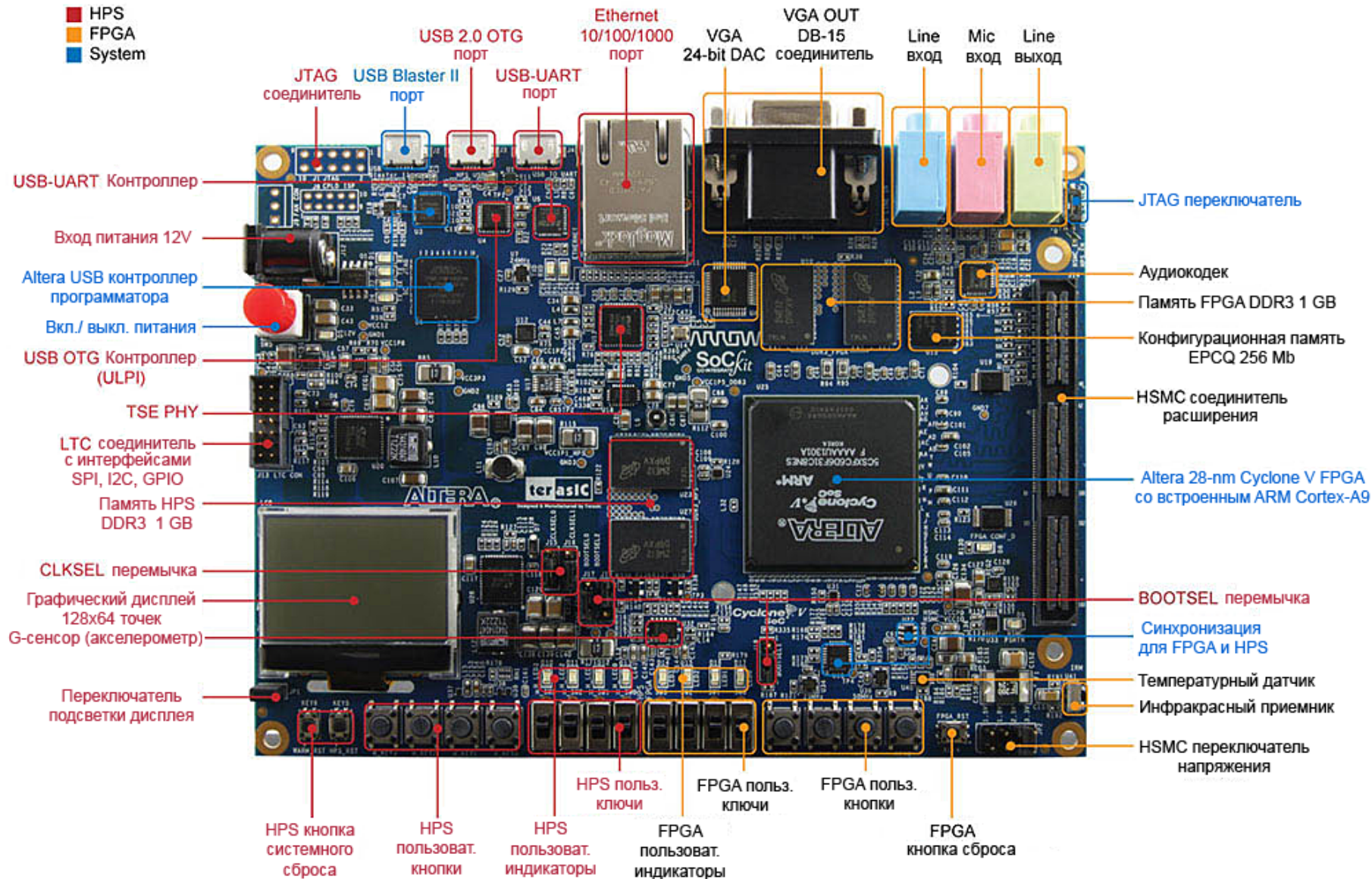


Отладочные платы на ПЛИС



Имеется в учебной лаборатории

Отладочные платы на ПЛИС и процессорах



Имеется на кафедре

Основные производители (разработчики) ПЛИС

Xilinx

Intel (Altera)

Microsemi (Actel)

Lattice Semiconductor

Achronix Semiconductor

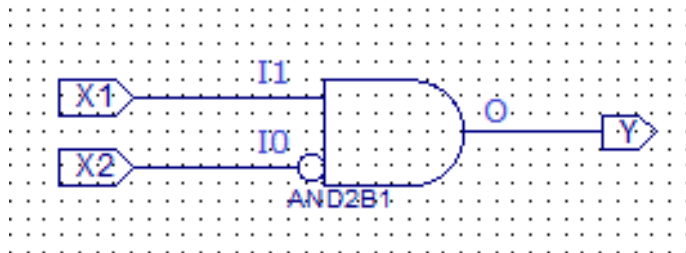
а также QuickLogic, Silego, Atmel, Cypress и др.

В России: Воронежский завод полупроводниковых приборов выпускает кристаллы по лицензии Altera.

Методы описания проектируемого устройства

Структурное (схемотехническое)

а) принципиальная схема с применением типовых библиотечных компонентов:



I1, I0, O — обозначение выводов компонента,
X1, X2, Y — обозначение внешних сигналов;

б) описание структуры и связей:

```
AND2B2 port map ( I1 => X1,  
                  I0 => X2,  
                  O  => Y );
```

Потоковое и поведенческое

Описание входных, выходных сигналов и функционирования системы на языке Hardware Description Language (HDL):

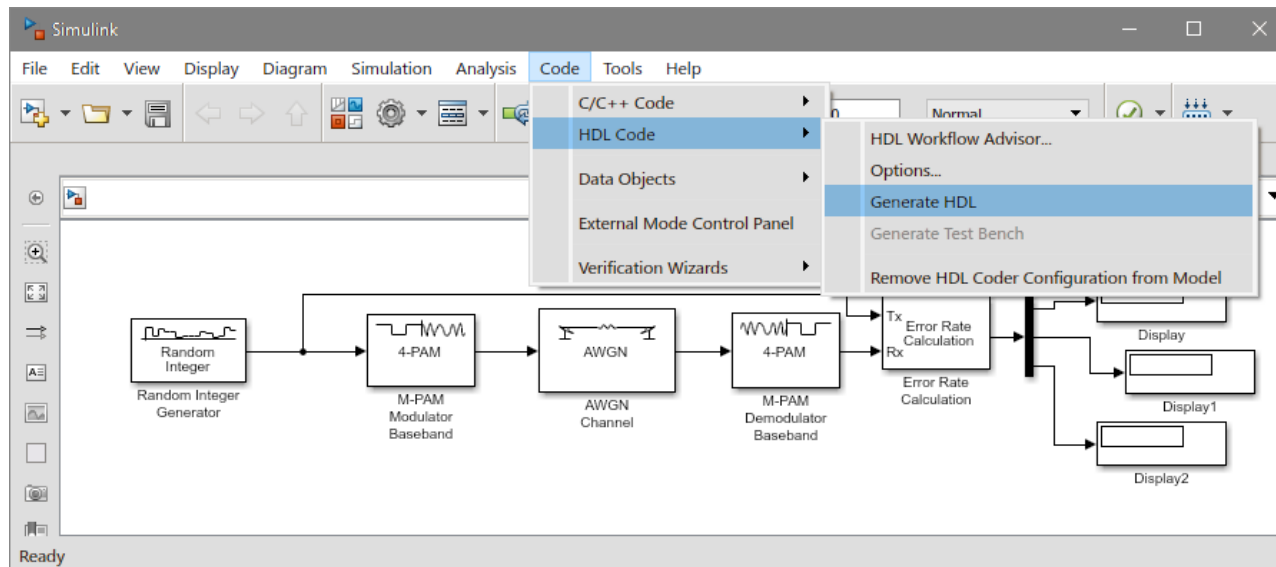
```
Y <= X1 and not X2;
```

Методы описания проектируемого устройства

Основные HDL:

- VHDL
- Verilog
- SystemVerilog
- SystemC

Создание описания из других сред моделирования



Основные элементы языка VHDL

Литература:

Тарасов И.Е. ПЛИС Xilinx. Языки описания аппаратуры VHDL и Verilog, САПР, приемы проектирования, 2019, с. 122–161],

VHDL: Справочное пособие по основам языка : учебное пособие / В. П. Бабак и др., 2010, с. 37–96]

VHDL — **V**HSIC (Very High-Speed Integrated Circuits) **H**ardware **D**escription **L**anguage

Объекты языка

| | |
|-----------------|--|
| constant | константа |
| variable | переменная |
| signal | сигнал — переносит информацию между компонентами |

Примеры записи констант

Числа (литералы):

123 45.6 7.8e-2 16#9ABC# 2#1000_0101# (знак _ необязателен)

Символы, строки символов, битовые строки:

'a' "string" "101100" b"1001111" x"4F"

Основные элементы языка VHDL

Основные типы данных

bit **integer** **std_logic** **std_logic_vector**

Тип **bit** принимает значения 0 и 1.

Тип **integer** описывает целые числа. Рекомендуется указывать диапазон:

```
signal x : integer range 0 to 255;
```

Тип **std_logic** является основным типом для представления сигналов.

Уровни:

- | | |
|--------------|--|
| 'U' , | Неинициализированный (сигнал, который еще не был задан). |
| 'X' , | Сильный неизвестный сигнал. |
| '0' , | Сильный лог. 0 |
| '1' , | Сильная лог. 1 |
| 'Z' , | Высокий импеданс (Z–состояние) |
| 'W' , | Слабый неизвестный сигнал |
| 'L' , | Слабый лог. 0 |
| 'H' , | Слабая лог. 1 |
| '–' | Уровень сигнала не имеет значения |

Основные элементы языка VHDL

Тип `std_logic_vector` — группа сигналов `std_logic`.

Примеры объявлений:

```
signal a1 : std_logic_vector (7 downto 0);  
или  
signal a2 : std_logic_vector (0 to 15);
```

Присвоение значения:

```
a1 <= "00000011";
```

Другие типы данных

array boolean character string real access time
перечислимый record signed unsigned natural

Основные элементы языка VHDL

Функции преобразования типов

`integer → std_logic_vector`

`<slv_sig> = CONV_STD_LOGIC_VECTOR(<int_sig>, <integer_size>);`

`std_logic_vector → integer`

`<int_sig> = CONV_INTEGER(<slv_sig>);`

`std_logic_vector → std_logic_vector (знаковое расширение)`

`<slv_sxt_sig> = SXT(<slv_sig>, <integer_size>);`

`std_logic_vector → std_logic_vector (дополнение нулями)`

`<slv_ext_sig> = EXT(<slv_sig>, <integer_size>);`

Основные элементы языка VHDL

Операторы

| Оператор | Название |
|---------------------------------|-------------------------------|
| Логические операторы | |
| and | И |
| nand | И-НЕ |
| or | ИЛИ |
| nor | ИЛИ-НЕ |
| xor | Исключающее ИЛИ |
| nxor | Исключающее ИЛИ-НЕ |
| not | НЕ |
| & | Объединение двоичных разрядов |
| Арифметические операторы | |
| + | Сложение |
| - | Вычитание, смена знака |
| * | Умножение |
| / | Деление |
| %, rem, mod | Остаток от деления |
| ** | Возведение в степень |
| abs | Абсолютное значение |
| Операторы отношения | |
| < | Меньше |
| > | Больше |
| <= | Меньше или равно |
| >= | Больше или равно |
| = | Равно |
| /= | Не равно |
| Операторы сдвига | |
| srl | Логический сдвиг вправо |
| sll | Логический сдвиг влево |
| sra | Арифметический сдвиг вправо |
| sla | Арифметический сдвиг влево |
| ror | Циклический сдвиг вправо |
| rol | Циклический сдвиг влево |

Основные элементы языка VHDL

Операторы присваивания

:= Присвоение значений переменным и начальная инициализация сигналов.

<= Присвоение значений сигналам.

Примеры:

```
variable index : integer := 0;  
y <= a or b;
```

Управляющие операторы

```
if...then...else  
case  
with...select  
for...loop  
while...loop  
process
```

Структурные операторы

```
entity  
architecture  
port map  
component  
generate
```

Структура модуля описания на VHDL

Описание интерфейса

Подключение библиотек

```
library ...  
use ...
```

Объявление интерфейса

```
entity ...  
port ...
```

Описание архитектуры и функционирования

```
architecture ...
```

Объявление внутренних сигналов, компонентов

```
signal ...  
component ...
```

Соединения компонентов

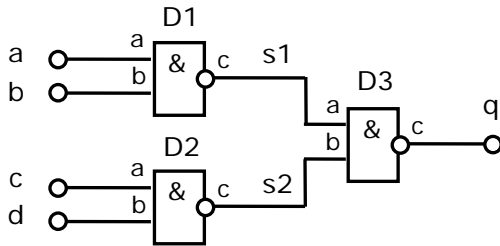
```
port map ...
```

Операции параллельного выполнения

```
Присвоение (потокосные)  
process ... (поведенческие)
```


Примеры описаний на языке VHDL

Структурное описание устройства



```
library IEEE;                                     -- Объявление библиотек
use IEEE.STD_LOGIC_1164.all;

-----

entity my_device is                                -- Начало описания объекта
    port (
        a, b, c, d : in std_logic; -- Входные сигналы порта
        q           : out std_logic -- Выходные сигналы порта
    );
end my_device;                                     -- Конец описания объекта

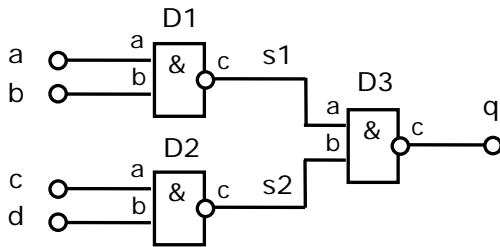
-----

architecture arch of my_device is                  -- Начало описания архитектуры

    signal s1, s2 : std_logic;                     -- Объявление внутренних сигналов

    component NAND2                                -- Объявление используемого компонента
    port (
        a, b : in std_logic;
        c     : out std_logic
    );
end component;
```

Структурное описание устройства (продолжение)



`begin`

`D1: NAND2 port map`

```
( a => a,  
  b => b,  
  c => s1  
);
```

`D2: NAND2 port map`

```
( a => c,  
  b => d,  
  c => s2  
);
```

`D3: NAND2 port map`

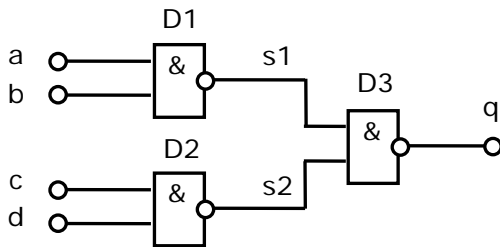
```
( a => s1,  
  b => s2,  
  c => q  
);
```

`end arch;`

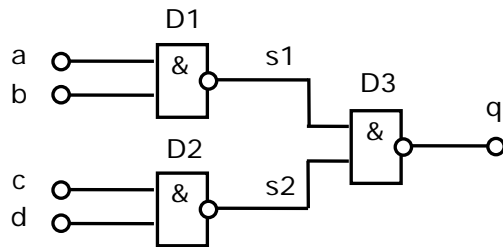
-- Начало описания структуры или
-- функционирования
-- Описание межкомпонентных связей:
-- имя вывода компонента => имя цепи

-- Конец описания архитектуры

Потоковое (поведенческое) описание устройства

[illegible]

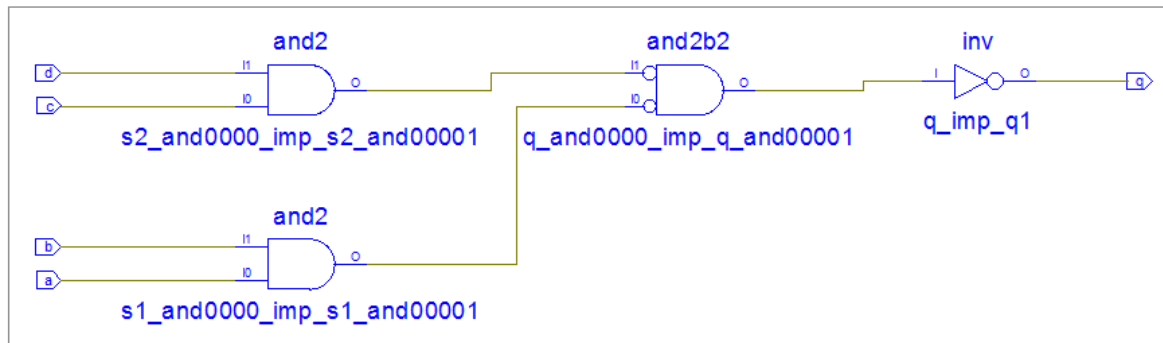
Потоковое (поведенческое) описание устройства



Результаты синтеза

Синтезированная структура (RTL Schematic)

RTL – Register transfer level



Синтезированная структура (Technology Schematic)

