

HDL Описание аппаратуры

Язык VHDL

К лекциям

«Интегрированные
информационно-
управляющие системы»

Кардаш Д.И.

KardashDI@narod.ru

Уфа, 2011

Структура описания VHDL

library Библиотека

use Элемент

use Элемент

entity Сущность

generic Параметры

port Порты

architecture
Архитектурное
тело

Определения

begin Действия

Операция

Операция

process Процесс

process Процесс

Определение типов

type

boolean

is (false,true);

-- ЛОГИЧЕСКИЙ ТИП

Определение типов

type

severity

is

(OKAY,NOTE,WARNING,
ERROR,FAILURE);

-- перечислимый тип

Модели данных

1. `constant` — константа;
2. `variable` — переменная;
3. `signal` — сигнал;
4. `file` — файлы;
5. `shared variable` —
задержанная переменная.

Примеры определения данных

```
signal x, y : std_logic;
```

```
variable temp : std_logic := '0';
```

```
constant gnd : bit := '0';
```

```
variable y : bit;
```

```
signal z : bit;
```

```
shared variable y : bit;
```

Оператор присваивания

- Для сигналов: $z \leq y$;
- Для всего остального: $y := \text{gnd}$.

Определение векторов

- type type_name is array (range) of element_type;
- type type_name is array (type range <>) of element_type; signal

type BYTE is array (7 downto 0) of BIT; -- объявили
8-ми битовый тип

signal BUT1 : BYTE; -- объявили 8-ми битовый
массив

std_logic_vector (7 downto 0);

std_logic_vector (0 to 7);

Пример определения параметров

<i>entity</i>	Сущность
---------------	----------

<i>generic</i>	Параметры
----------------	-----------

<i>port</i>	Порты
-------------	-------

Например:

```
generic ( width : integer := 8 );
```

Типы портов

- in - входной порт (только для чтения значения сигнала);
- out - выходной порт (только для записи значения сигнала);
- inout - двунаправленный порт (чтение и запись);
- buffer - выходной порт, значение которого можно считывать;
- linkage - двунаправленный порт с ограниченным использованием (практически не используется).

Определение портов

```
port (  
a, b : in std_logic_vector(0 to 3);  
s : out std_logic_vector(0 to 3);  
c : out std_logic -- Перенос  
);
```

Сущность

entity

Сущность

generic

Параметры

port

Порты

```
entity lfsr is  
    generic ();  
    port ();  
end lfsr;
```

Архитектура

architecture

Архитектурное
тело

Определения

begin

Действия

Операция

Операция

process

Процесс

process

Процесс

architecture

NAME_ARCHITECTURE of
NAME_ENTITY is

-- Здесь содержатся:

описания типов данных;
функции и процедуры;
компоненты более низкого
уровня иерархии; описания
сигналов и глобальных
переменных.

begin

-- Здесь содержатся исполняемые
операторы.

end NAME_ARCHITECTURE;

Пример описания архитектуры

architecture **ИмяАрх** of **ИмяПортов** is

variable temp : std_logic := '0';

begin

 mrand_out(3) <= rand_in(7);

 mrand_out(2) <= rand_in(5) AND rand_in(4);

 mrand_out(1) <= rand_in(3) AND rand_in(2);

 mrand_out(0) <= rand_in(1) AND rand_in(0);

end ИмяАрх;

Операции VHDL

- логические операции (and, or, nand, nor, xor, xnor);
- операции сравнения (=, /=, <, <=, >, >=);
- операции сдвига (sll, srl, sla, sra, rol, ror);
- операции сложения (+, -, &);
- операции смены знака числа (+, -);
- умножение деление (*, /, mod, rem);
- прочие (**, abs, not).

Операторы VHDL

- *Последовательные и параллельные.*
- *Синтезируемые и внедренные.*
- *Простые и составные.*
- *Метки L1 : for i in 0 to 31 loop.*

Операции VHDL

- логические операции (and, or, nand, nor, xor, xnor);
- операции сравнения (=, /=, <, <=, >, >=);
- операции сдвига (sll, srl, sla, sra, rol, ror);
- операции сложения (+, -, &);
- операции смены знака числа (+, -);
- умножение деление (*, /, mod, rem);
- прочие (**, abs, not).

УСЛОВИЯ

if *условие1* then

выполнение условия1

elsif *условие2* then

выполнение условия2

else

невыполнение условия

end if;

ЦИКЛЫ

```
for i in 5 downto 0 loop  
    операции в цикле  
end loop;
```

```
for i in 0 to 5 loop  
    операции в цикле  
end loop;
```

ЦИКЛЫ

```
count : while i<=N loop
```

операции в цикле, счетная
переменная вычисляется

```
end loop count;
```

Выход из цикла.

```
L1 : ...
```

```
exit L1
```

Множественный выбор

case переменная is

when `1` => out:=1;

when `0` => out:=0;

end case;

Оператор ожидания

- WAIT on X,Y (список чувствительности) until ($Z=0$) for 100ns;
- WAIT on X,Y until ($Z=0$);
- WAIT on X,Y;
- WAIT for 100ns;

Параллельный оператор процесса

LABEL: process (лист чувствительности)

-- область деклараций

begin

--VHDL операторы

end process;

Например:

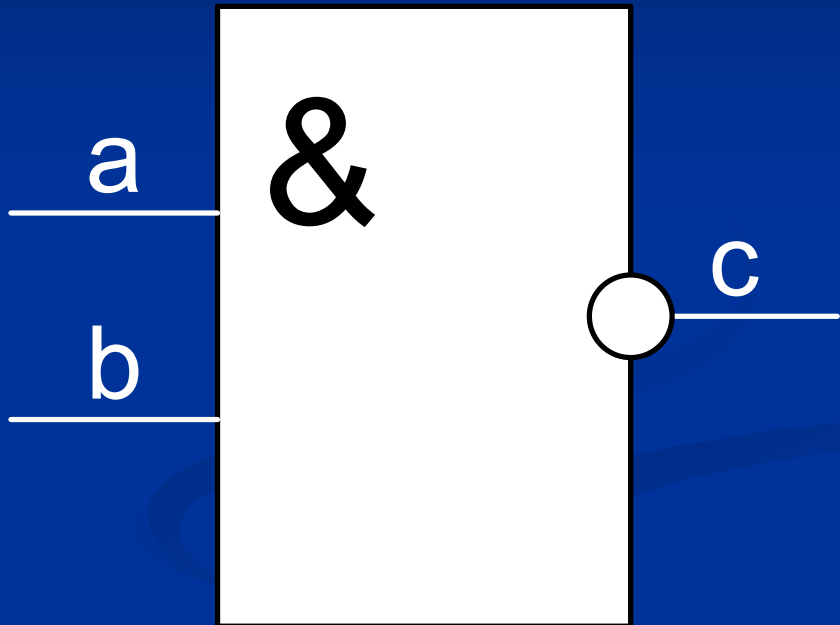
process(X,Y,Z)

исполняемые действия

end process;

Реализация элемента 2И-НЕ

```
library ieee;  
use ieee.std_logic_1164.all;  
-----  
entity NAND2 is  
port (  
  a, b : in std_logic;  
  c : out std_logic  
);  
end NAND2;  
-----  
architecture aaa of NAND2 is  
begin  
  c <= not (a and b);  
end aaa;
```



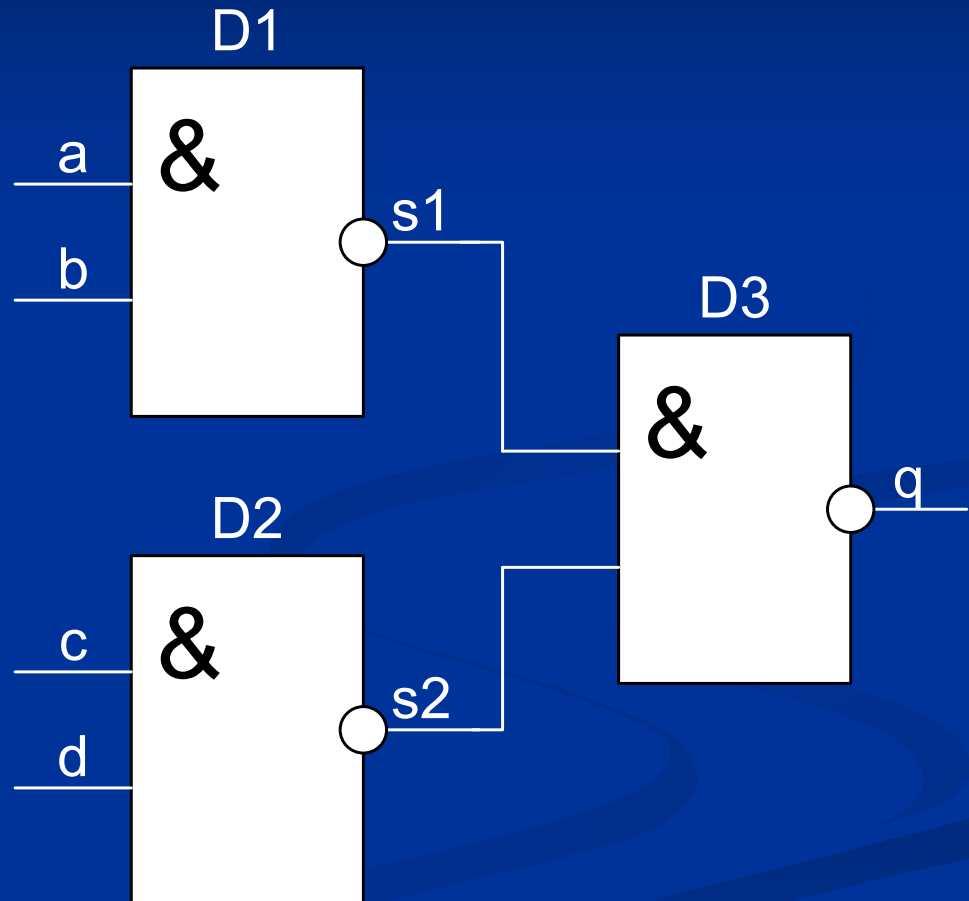
Описание компонента

```
component NAND2
    port ( описание портов );
end component;
```

```
library ieee;
use ieee.std_logic_1164.all;
-----
entity NAND2 is
port (
a, b : in std_logic;
c : out std_logic
);
end NAND2;
-----
architecture aaa of NAND2 is
begin
c <= not (a and b);
end aaa;
```

Структурное описание схемы

```
library ieee;  
use ieee.std_logic_1164.all;  
-----  
entity NAND2X3 is  
port (  
  a, b, c, d : in std_logic;  
  q : out std_logic  
);  
end NAND2X3;  
-----
```



Структурное описание схемы

architecture bbb of NAND2X3 is
signal s1, s2 : std_logic;
component NAND2

port (
a, b : in std_logic;
c : out std_logic);
end component;

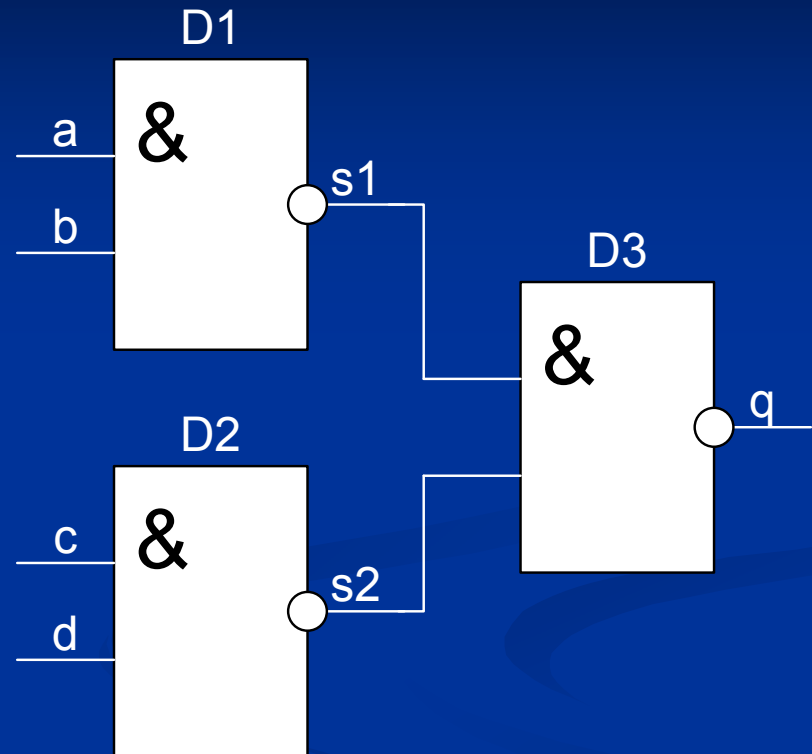
begin

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 bbb;



Поведенческое описание схемы

entity NAND2X3 is

port (

a, b, c, d : in std_logic;

q : out std_logic

);

end NAND2X3;

architecture ccc of NAND2X3 is

signal s1, s2 : std_logic;

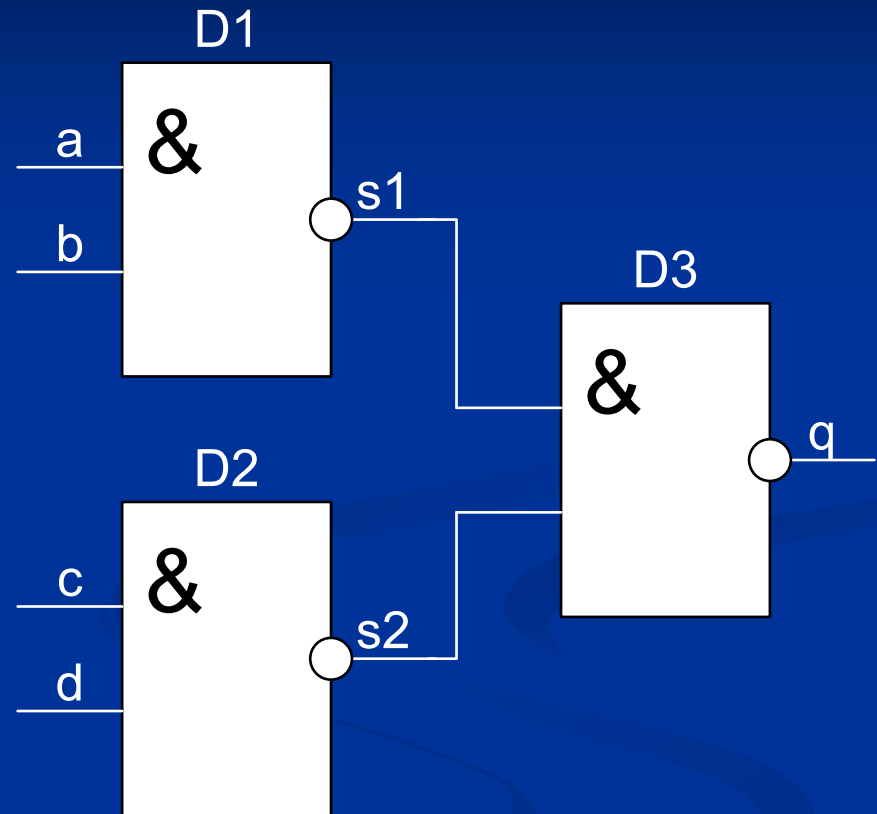
begin

s1 <= not (a and b);

s2 <= not (c and d);

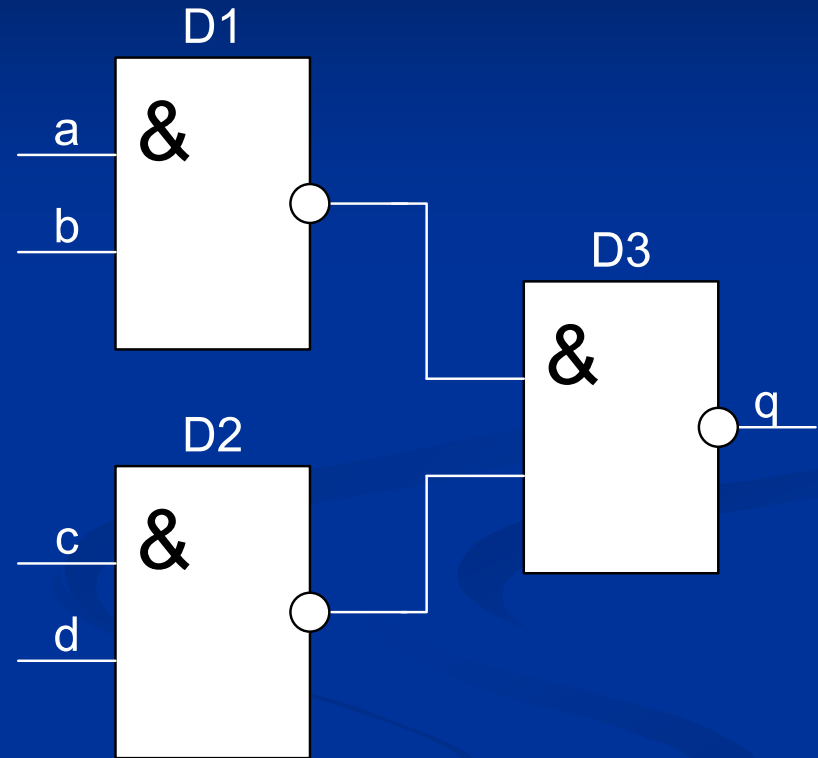
q <= not (s1 and s2);

end ccc;



Поведенческое описание II

```
entity NAND2X3 is
port (
a, b, c, d : in std_logic;
q : out std_logic
);
end NAND2X3;
architecture ddd of NAND2X3 is
begin
q <= (a and b) or (c and d);
end ddd;
```



Типы атрибутов

Атрибуты - это характеристики объектов VHDL.

Атрибуты бывают:

- predetermined;
- user-defined.

Predetermined attributes are:

- signal attributes;
- array attributes;
- type attributes.

Атрибуты сигналов

- $S'active$ – TRUE, если было присвоение, но текущее значение еще прежнее
- $S'delayed(t)$ – значение сигнала, существовавшее на время t перед вычислением данного атрибута
- $S'event$ – TRUE, если происходит изменение сигнала
- $S'last_active$ – время от последнего присвоения значения сигналу до момента вычисления атрибута $S'last_event$ Время от последнего изменения сигнала до момента вычисления атрибута
- $S'last_value$ – последнее присвоенное сигналу значение $S'stable(t)$ TRUE, если не происходило изменение сигнала в течение времени t
- $S'transaction$ – TRUE, если происходит очередное присвоение значения сигналу
- $S'quiet$ – FALSE, если было присвоение, но текущее значение еще прежнее

Описание функций

```
function rising_edge (signal s : std_ulogic) return boolean is  
begin  
  return (s'event and s = '1');  
end rising_edge;
```

```
function falling_edge (signal s : std_ulogic) return boolean is  
begin  
  return (s'event and s = '0');  
end falling_edge;
```


Описание процедур

```
procedure Name_Proc (signal s : std_logic);  
begin  
    действия  
return;  
end Name_Proc;
```

Отличия от функций:

Процедура не возвращает данных