



Какво представлява VHDL?

VHDL – Език за описание на електронни системи.

Описание на структура

Описание на поведение

Документиране

Симулация

Синтез

Технологично независим

История

VHDL = VHSIC Hardware Description Language

VHSIC = Very High Speed Integrated Circuits (част от програмата "Star wars")

на Министерство на отбраната на САЩ

IEEE Std. 1076 – 1987 (VHDL – 87)

IEEE Std. 1076 – 1993 (VHDL – 93)

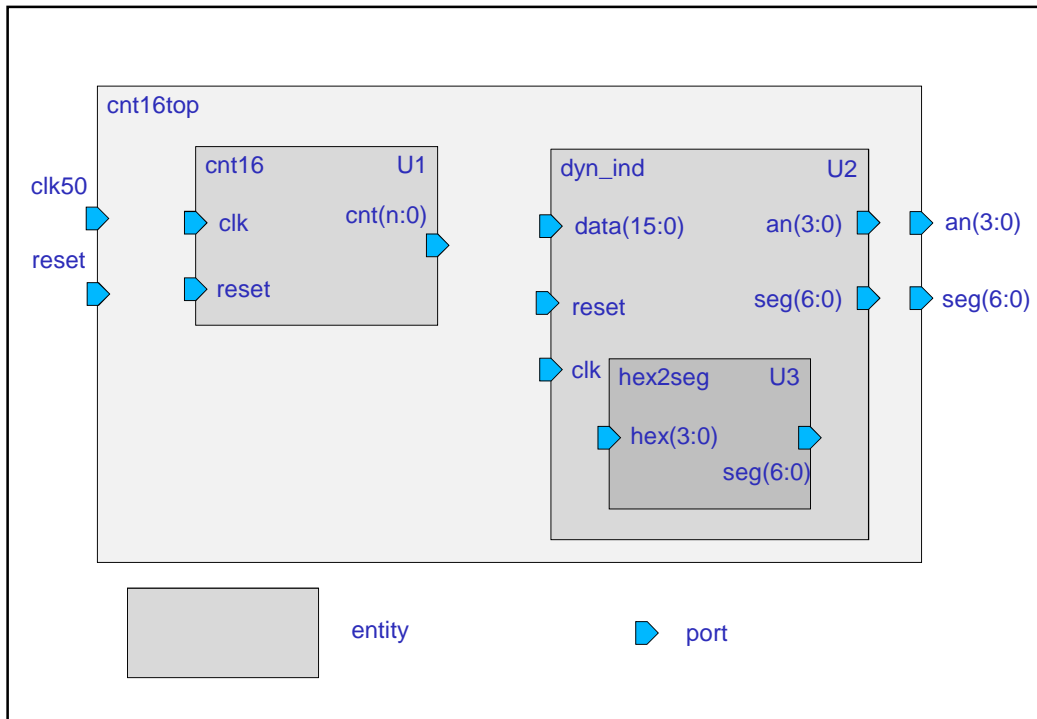
Алтернативни HDL – Verilog, ...

Йерархия

Entity

Port

Architecture



Entity and Ports

```

entity cnt16 is
  generic (n: integer);
  port ( clk : in  std_logic;
        reset : in  std_logic;
        cnt : out std_logic_vector (n-1 downto 0));
end cnt16;

```

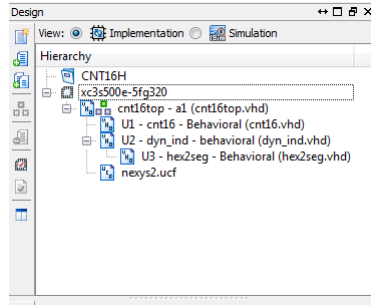
Architecture

```
architecture Behavioral of cnt16 is
    signal cnt_i : unsigned(n-1 downto 0);
begin
    process(clk, reset)
    begin
        if( reset = '1') then
            cnt_i <= (others=>'0');
        elsif rising_edge(clk) then
            cnt_i <= cnt_i + 1;
        end if;
    end process;
    cnt <= std_logic_vector(cnt_i);
end Behavioral;
```

Instantiation

```
U1: entity work.cnt16(behavioral)
    generic map (n => 16)
    port map(
        clk => clk,
        reset => reset,
        cnt => cnt
    );

U2: entity work.dyn_ind(behavioral)
    port map(
        clk => clk_ind,
        reset => reset,
        data => cnt,
        seg => seg,
        an => an
    );
```



See cnt16 project

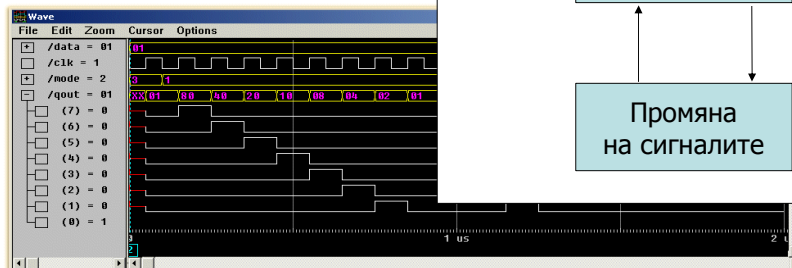
http://lark.tu-sofia.bg/mpis/labs/lab_intro/cnt16vhd/

VHDL Симулация

Събитие – промяна на сигнал

Симулационен цикъл

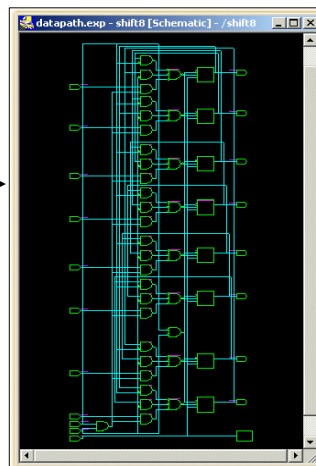
Делта – условна единица за време



Синтез

```
process begin
wait until (rising_edge(clk));
case mode is
when "01" =>
    tmp <= tmp(0) & tmp(7 downto 1);
when "10" =>
    tmp <= tmp(6 downto 0) & tmp(7);
when "11" =>
    tmp <= data;
when others =>
    null;
end case;
end process;
```

СИНТЕЗ



Не всеки VHDL модел е синтезируем!

Сигнали

- Свързващи проводници
- Опашки от данни
- Тип – набор от допустими стойности
- bit, integer, std_logic ...

Bit: 0,1

Std_logic: 0,1,L,H,X,W,Z,U,-

Портове

Портове – сигнали с указана посока

in – драйверът е извън проектната единица.

out – драйверът е в пр. единица. Сигналят не може да бъде прочетен в пр. единица!

inout - драйверът може бъде както в, така и извън пр. единица.

Типове данни

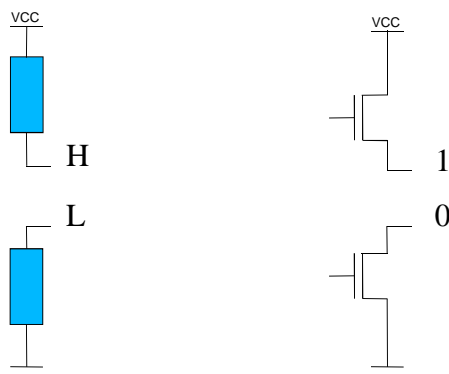
bit, std_logic, std_logic_vector,
integer, unsigned, signed, ...

STD_LOGIC

Типът STD_LOGIC е дефиниран в пакета IEEE.std_logic_1164

```
TYPE std_ulogic IS ( 'U', -- Uninitialized
                    'X', -- Forcing Unknown
                    '0', -- Forcing 0
                    '1', -- Forcing 1
                    'Z', -- High Impedance
                    'W', -- Weak Unknown
                    'L', -- Weak 0
                    'H', -- Weak 1
                    '-' -- Don't care
                    );
```

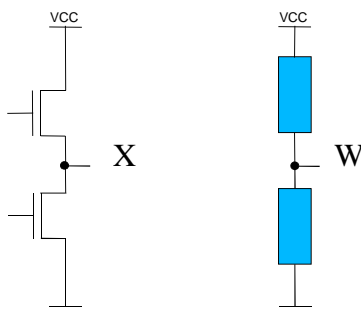
STD_LOGIC



L – слаба нула – пасивен път
(резистор) към земя
H – слаба единица – пасивен път
(резистор) към VCC

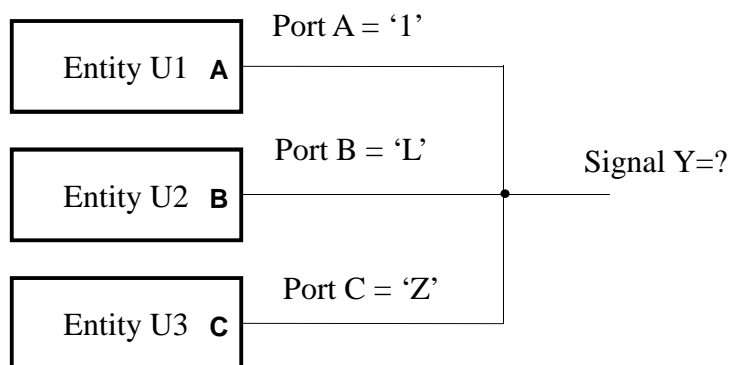
0 – силна нула – активен път към земя
1 – силна единица – активен път към VCC

STD_LOGIC



X – силно неопределено – активни пътища към VCC и земя
W – слабо неопределено – пасивни пътища към VCC и земя

STD_LOGIC



Libraries, numerical data types

IEEE	VHDL'87/VHDL'93	SYNOPSYS proprietary	draft math
std_logic_1164.vhd (ieee)	standard.vhd (std)	std_logic_arith.vhd (ieee)	mathpack.vhd
numeric_std.vhd (ieee)	textio.vhd (std)	std_logic_misc.vhd (ieee)	
numeric_bit.vhd (ieee)		std_logic_signed.vhd (ieee)	
		std_logic_unsigned.vhd (ieee)	
		std_logic_textio.vhd (ieee)	
		attributes.vhd (synopsys)	

Types in Binary Arithmetic Operations			
		numeric_std	std_logic_arith
Argument 1	Argument 2	Result	
unsigned	unsigned	unsigned	unsigned/std_logic_vector
unsigned	integer	unsigned	unsigned/std_logic_vector
integer	unsigned	unsigned	unsigned/std_logic_vector
signed	signed	signed	signed/std_logic_vector
signed	integer	signed	signed/std_logic_vector
integer	signed	signed	signed/std_logic_vector

	numeric_std	std_logic_arith
Type Conversion		
std_logic_vector -> unsigned	unsigned(arg)	unsigned(arg)
std_logic_vector -> signed	signed(arg)	signed(arg)
unsigned -> std_logic_vector	std_logic_vector(arg)	std_logic_vector(arg)
signed -> std_logic_vector	std_logic_vector(arg)	std_logic_vector(arg)
integer -> unsigned	to_unsigned(arg, size)	conv_unsigned(arg, size)
integer -> signed	to_signed(arg, size)	conv_signed(arg, size)
unsigned -> integer	to_integer(arg)	conv_integer(arg)
signed -> integer	to_integer(arg)	conv_integer(arg)
integer -> std_logic_vector	integer -> unsigned/signed -> std_logic_vector	
std_logic_vector -> integer	std_logic_vector -> unsigned/signed -> integer	
unsigned + unsigned -> std_logic_vector	std_logic_vector(arg1 + arg2)	arg1 + arg2
signed + signed -> std_logic_vector	std_logic_vector(arg1 + arg2)	arg1 + arg2
Resizing		
unsigned	resize(arg, size)	conv_unsigned(arg, size)
signed	resize(arg, size)	conv_signed(arg, size)

Организация на VHDL проект

VHDL Проект

Пакети, библиотеки:

съдържат декларации на типове, процедури

Интерфейсна част (entity):

Описание на портове и
общи (generic) параметри

Архитектура (architecture):

Описва реализацията на
проектната единица

Проектна единица

Архитектура

Архитектура

Декларации на:

типове, процедури, сигнали, константи

Тяло – съдържа паралелни оператори:

- Паралелни сигнални присвоявания;
- Процеси;
- Включване на компоненти;
- Извикване на процедури.

Процеси

Декларации на:

- променливи
- локални типове, процедури

Тяло – съдържа последователни оператори:

- Сигнални присвоявания;
- Присвоявания на променливи;
- if, case, loop;
- Извикване на процедури;
- wait;
- null.

Разработване на VHDL Модел

Спецификация

Интерфейс (entity)

Архитектура (architecture)

Спецификация

Да се проектира VHDL модел реализиращ функцията зададена със следната таблица на истинност:

a	b	c	y
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

```
library IEEE;  
use IEEE.STD_LOGIC_1164.ALL;
```

```
entity fun is  
  Port ( a : in STD_LOGIC;  
        b : in STD_LOGIC;  
        c : in STD_LOGIC;  
        y : out STD_LOGIC);  
end fun;
```

```
architecture Behavioral of fun is
```

```
begin
```

```
  y <= (not a and not b and not c) or  
       (not a and not b and c) or  
       (not a and b and c) or  
       (a and not b and c) or  
       (a and b and c);
```

```
end Behavioral;
```

Entity & Architecture

Синтез

