

Практика
Описание комбинационных
устройств на VHDL

Пример 1. Проверка на четность



```
library ieee;
use ieee.std_logic_1164.all;

entity parity_check is
    port(
        a: in std_logic_vector(7 downto 0);
        x: out std_logic
    );
end parity_check;

architecture cond_arch of parity_check is
begin
    x <= '1' when (a(0) = '0') else
        '0';
end cond_arch;
```

| input | output |
|-------|--------|
| s | x |
| 00 | a |
| 01 | b |
| 10 | c |
| 11 | d |

Пример 2. Проверка на четность (if)



```
architecture cond_arch of parity_check is
begin
  process(a)
  begin
    If (a(0) = '0') then
      x <= '1';
    else
      x <= '0';
    end if;
  end process;
end cond_arch;
```

| input | output |
|-------|--------|
| s | x |
| 00 | a |
| 01 | b |
| 10 | c |
| 11 | d |

Пример 3. Делимость на 4



```
architecture cond_arch of check_4 is
begin
  process(a)
  begin
    If ((a(0) = '0') and (a(1) = '0')) then
      x <= '1';
    else
      x <= '0';
    end if;
  end process;
end cond_arch;
```

| input s | output x |
|------------|-------------|
| 00 | a |
| 01 | b |
| 10 | c |
| 11 | d |

Пример 4. Делимость на 4 без and

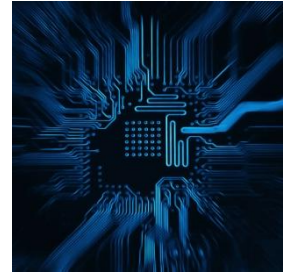


```
architecture cond_arch of check_4 is
begin
  process(a)
  begin
    If (a(1 downto 0) = "00") then
      x <= '1';
    else
      x <= '0';
    end if;
  end process;
end cond_arch;
```

| input s | output x |
|------------|-------------|
| 00 | a |
| 01 | b |
| 10 | c |
| 11 | d |

Пример 5. Дектор кодов

0101 – 11; 1010 – 10; 11111 – 01; x - 00



```
architecture cond_arch of parity_check is  
begin  
  process(a)  
  begin  
    if (a(3 downto 0) = "0101") then  
      x <= "11";  
    elsif (a(3 downto 0) = "1010") then  
      x <= "10";  
    elsif (a(4 downto 0) = "11111") then  
      x <= "01";  
    else  
      x <= "00";  
    end if;  
  end process;  
end cond_arch;
```

Пример 6. Сумма двух двоичных четырехразрядных чисел

```
library ieee;
use ieee.std_logic_1164.all;
entity adder_4_2 is
    port(
        a,b: in std_logic_vector(3 downto 0);
        x: out std_logic_vector(4 downto 0)
    );
end adder_4_2;

architecture test1 of adder_4_2 is
    signal s: std_logic_vector (3 downto 0);
begin
    x(0) <= a(0) xor b(0);
    s(0) <= a(0) and b(0);
    x(1) <= (a(1) xor b(1)) xor s(0);
    s(1) <= ((a(1) xor b(1)) and s(0)) or (a(1) and b(1));
    x(2) <=
    s(2) <=
    x(3) <=
    x(4) <=
end test1;
```

