

# VHDL II

Tom Kelliher, CS 220

Mar. 13, 2006

## 1 Administrivia

### Announcements

### Assignment

Read 6.1–2.

Assignment due Mar. 27.

### From Last Time

Structural VHDL.

### Outline

1. Dataflow VHDL.
2. Hierarchical VHDL.
3. Behavioral VHDL.
4. Class practice.

## Coming Up

Introduction to sequential circuits. Latches.

## 2 Dataflow VHDL

Last time's EXOR3:

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

entity EXOR3 is
    port
    (
        i : in std_logic_vector (2 downto 0),
        o : out std_logic
    );
end EXOR3;

architecture dataflow of EXOR3 is

    signal i2_n, i1_n, i0_n : std_logic;
    signal m1, m2, m3, m4 : std_logic;      -- Minterms.

begin
    i2_n <= not i(2);
    i1_n <= not i(1);
    i0_n <= not i(0);

    m1 <= i2_n and i1_n and i(0);
    m2 <= i2_n and i(1) and i0_n;
    m3 <= i(2) and i1_n and i0_n;
    m4 <= i(2) and i(1) and i(0);

    o <= m1 or m2 or m3 or m4;
end dataflow;

architecture function_table of EXOR3 is
begin
    with i select
```

```

        o <= '0' when "000",
            '1' when "001",
            '1' when "010",
            '0' when "011",
            '1' when "100",
            '0' when "101",
            '0' when "110",
            '1' when "111",
            'X' when others;
end function_table;

```

new VHDL elements: vectors and downto (also “to”). not, and, or. Concurrent assignment. With-Select, when, others.

### 3 Hierarchical VHDL

Use of components. Full binary adder constructed from two half-adders example:

```

library ieee;
use ieee.std_logic_1164.all;

entity ha is
    port
    (
        a, b : in std_logic;
        c, s : out std_logic
    );
end ha;

architecture dataflow of ha is
begin
    s <= a xor b;
    c <= a and b;
end dataflow;

library ieee;
use ieee.std_logic_1164.all;

entity fa is

```

```

    port
    (
        a, b c_i : in std_logic;
        c_o, s    : out std_logic
    );
end fa;

architecture mixed of fa is

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

    signal hs, hc, tc : std_logic;

begin
    ha1 : ha port map (a, b, hc, hs);
    ha2 : ha port map (hs, c_i, tc, s);

    c_o <= hc or tc;
end mixed;

```

Nothing really new — we’ve seen components already.

## 4 Behavioral VHDL

High level VHDL. Four bit adder:

```

library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;

entity adder4 is
    port
    (a, b : in std_logic_vector (3 downto 0);
    c0    : in std_logic;

```

```

        s      : out std_logic_vector (3 downto 0);
        c4     : out std_logic
    );
end adder4;

architecture behavioral of adder4 is

    signal sum : std_logic_vector (4 downto 0);

begin
    sum <= ('0' & a) + ('0' & b) + ("0000" & c0);
    c4 <= sum(4);
    s <= sum(3 downto 0);
end behavioral;

```

Note use of “high level” operators: +, &.

## 5 Class Practice

Write dataflow (includes function table) VHDL for a 4-1 mux.

Using the fa component, design a hierarchical four-bit adder.

Write dataflow VHDL for an eight-input priority encoder.