

VHDL Vectors and Test Benches

Tom Kelliher, CS 240

Mar. 23, 2012

1 Administrivia

Announcements

Assignment

From Last Time

Structural VHDL lab.

Outline

1. VHDL vectors.
2. Test bench styles.
3. Class practice.

Coming Up

2 VHDL Vectors

VHDL vectors are used to group signals together (think buses):

```

entity mux41 is
    Port ( d : in  STD_LOGIC_VECTOR (3 downto 0);
          s : in  STD_LOGIC_VECTOR (1 downto 0);
          x : out STD_LOGIC);
end mux41;

architecture structural of mux41 is

    ...

begin

m0: mux21 port map(d(1), d(0), s(0), t0);
m1: mux21 port map(d(3), d(2), s(0), t1);
m2: mux21 port map(t1, t0, s(1), x);

end structural;

library IEEE, lcdf_vhdl;
use IEEE.STD_LOGIC_1164.ALL, lcdf_vhdl.func_prims.all;

entity mux21 is
    Port ( d1 : in  STD_LOGIC;
          d0 : in  STD_LOGIC;
          s0 : in  STD_LOGIC;
          x : out  STD_LOGIC);
end mux21;

architecture structural of mux21 is

    ...

begin

    ...

end structural;

```

Note:

- Declaration syntax, type, and the use of `downto`.
- Use of `(` and `)` to “index” a specific bit of the vector.

- That separate `library` and `use` statements must precede each `entity` declaration.
- These uses are common:

```

signal instr    : std_logic_vector (31 downto 0);
signal opcode   : std_logic_vector (5  downto 0);
signal alu_a_in : std_logic_vector (31 downto 0);

...

opcode <= instr (31 downto 26);
alu_a_in <= X"0000" & instr (15 downto 0);

```

3 Test Bench Styles

“Signal generator” style:

```

LIBRARY ieee;
USE ieee.std_logic_1164.ALL;

ENTITY tb01 IS
END tb01;

ARCHITECTURE behavior OF tb01 IS

    -- Component Declaration for the Unit Under Test (UUT)

    COMPONENT mux21
    PORT(
        d1 : IN  std_logic;
        d0 : IN  std_logic;
        s0 : IN  std_logic;
        x  : OUT std_logic
    );
    END COMPONENT;

    --Inputs
    signal d1 : std_logic := '0';
    signal d0 : std_logic := '0';

```

```

signal s0 : std_logic := '0';

--Outputs
signal x : std_logic;

constant period : time := 10 ns;

BEGIN

-- Instantiate the Unit Under Test (UUT)
 uut: mux21 PORT MAP (
    d1 => d1,
    d0 => d0,
    s0 => s0,
    x => x
  );

-- Clock process definitions
s0_process :process
begin
  s0 <= '0';
  wait for period;
  s0 <= '1';
  wait for period;
end process;

d0_process :process
begin
  d0 <= '0';
  wait for 2 * period;
  d0 <= '1';
  wait for 2 * period;
end process;

d1_process :process
begin
  d1 <= '0';
  wait for 4 * period;
  d1 <= '1';
  wait for 4 * period;
end process;

END;
```

Notes:

- Good for exhaustive testing.
- Necessary for clock signals.

“Transitions” style:

```
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;

ENTITY tb00 IS
END tb00;

ARCHITECTURE behavior OF tb00 IS

    -- Component Declaration for the Unit Under Test (UUT)

    COMPONENT mux41
    PORT(
        d : IN  std_logic_vector(3 downto 0);
        s : IN  std_logic_vector(1 downto 0);
        x : OUT std_logic
    );
    END COMPONENT;

    --Inputs
    signal d : std_logic_vector(3 downto 0) := (others => '0');
    signal s : std_logic_vector(1 downto 0) := (others => '0');

    --Outputs
    signal x : std_logic;

    constant period : time := 10 ns;

BEGIN

    -- Instantiate the Unit Under Test (UUT)
    uut: mux41 PORT MAP (
        d => d,
        s => s,
```

```

        x => x
    );

-- Stimulus process
stim_proc: process
begin

    d <= "1110";
    s <= "00";

    wait for period;

    d <= "0001";
    s <= "00";

    wait for period;

    d <= "1101";
    s <= "01";

    wait for period;

    d <= "0010";
    s <= "01";

    wait for period;

    d <= "1011";
    s <= "10";

    wait for period;

    d <= "0100";
    s <= "10";

    wait for period;

    d <= "0111";
    s <= "11";

    wait for period;

    d <= "1000";
    s <= "11";

```

```

    wait for period;

    -- Wait indefinitely.
    wait;

end process;

END;

```

Notes:

- Good when one has a specific set of test vectors.
- May be combined with the signal generator style for sequential circuits, but waits must be coordinated with the clock signal's period.

I.e., the stimulus process will utilize `wait for period;` and the clock process will utilize `wait for period/2;;`

```

clk_process :process
begin
    clk <= '0';
    wait for period/2;
    clk <= '1';
    wait for period/2;
end process;

stim_proc:process
begin

    -- Allow 10 clock cycles for circuit to initialize.
    reset_n <= '0';
    wait for 10*period;

    reset_n <= '1';
    enbl <= '1';
    d <= "0101";
    wait for period;

    enbl <= '0';
    wait for period;

```

```
    ...  
    wait;  
end process;
```

4 Class Practice

Using your mux21 from the last class, implement, synthesize, and simulate a mux41. Do not perform exhaustive testing.