# Logic and Computer Design Fundamentals VHDL

#### Part 2 – Chapter 5 – Behavioral and Hierarchical Description

#### **Charles Kime & Thomas Kaminski**

© 2004 Pearson Education, Inc. <u>Terms of Use</u> (Hyperlinks are active in View Show mode)

## Overview

- Part 1 VHDL Basics and Types of Descriptions
- Part 2 Behavioral and Hierarchical Description
  - VHDL Hierarchy
    - Entities
    - Entity Instantiation
  - Behavioral Descriptions
    - Using:
      - Assignment statements
      - Higher-level operators
- Part 3 Finite State Machines
- Part 4 Registers and Counters
- Part 5 Algorithmic State Machine Example: Binary Multiplier

```
-- TOP LEVEL OF HIERARCHY
-- 4-bit adder-subtractor
library ieee;
use ieee.std_logic_1164.all;
entity addsub is
port(A,B: in std logic vector(3 downto 0);
      sub: in std logic;
      R: out std logic vector(3 downto 0));
end addsub;
architecture hierarchy of addsub is
 component add
  port(X,Y: in std logic vector(3 downto 0);
       C in: in std logic;
       S: out std logic vector(3 downto 0));
 end component;
```

Logic and Computer Design Fundamentals PowerPoint<sup>®</sup> Slides © 2004 Pearson Education, Inc.

```
component M1comp
 port(data in: in std logic vector(3 downto 0);
       comp: in std logic;
       data out: out std logic vector(3 downto 0));
 end component;
signal data out: std logic vector(3 downto 0);
begin
al: add
 port map(X => A, Y => data out, C in => sub, S =>
 R);
m1: M1comp
  port map(data in => B, data out => data out, comp
 => sub);
end hierarchy;
```

Logic and Computer Dealon Fundamentals PowerPoint<sup>®</sup> Slides © 2004 Pearson Education, Inc.

VHDL - Part 2 4

--add ENTITY AS INSTANTIATED ABOVE

library ieee;

```
use ieee.std_logic_1164.all, ieee.std_logic_unsigned.all;
```

-- Package "unsigned" required for type conversion to

```
-- perform + on std_logic signals instead of on integers
entity add is
```

port(X,Y: in std\_logic\_vector(3 downto 0);

C\_in: in std\_logic;

```
S: out std_logic_vector(3 downto 0));
```

end add;

architecture behavioral of add is

begin

 $S \le X + Y + ("000" \& C in);$ 

end behavioral;

Logic and Computer Dealon Fundamentals PowerPoint<sup>®</sup> Slides © 2004 Pearson Education, Inc.

```
--M1comp ENTITY AS INSTANTIATED ABOVE
library ieee;
use ieee.std logic 1164.all;
entity M1comp is
 port(data in: in std logic vector(3 downto 0);
       comp: in std logic;
       data out: out std logic vector(3 downto 0));
end M1comp;
architecture behavioral of M1comp is
begin
 data out <= (comp & comp & comp & comp) xor
  data in;
end behavioral;
```

Logic and Computer Dealgn Fundamentals PowerPoint<sup>®</sup> Slides © 2004 Pearson Education, Inc.

# **Terms of Use**

- © 2004 by Pearson Education, Inc. All rights reserved.
- The following terms of use apply in addition to the standard Pearson Education <u>Legal Notice</u>.
- Permission is given to incorporate these materials into classroom presentations and handouts only to instructors adopting Logic and Computer Design Fundamentals as the course text.
- Permission is granted to the instructors adopting the book to post these materials on a protected website or protected ftp site in original or modified form. All other website or ftp postings, including those offering the materials for a fee, are prohibited.
- You may not remove or in any way alter this Terms of Use notice or any trademark, copyright, or other proprietary notice, including the copyright watermark on each slide.
- Return to Title Page