# Logic and Computer Design Fundamentals VHDL

Part 5 – Chapter 8 – Algorithmic State Machine Example: Binary Multiplier

#### **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
- Part 3 Finite State Machines
- Part 4 Registers and Counters
- Part 5 Algorithmic State Machine Example: Binary Multiplier
  - Conversion of ASM into VHDL Description
    - Decomposition
      - Sequencing Synchronous process for FSM state, combinational process for FSM next state
      - Register transfers -synchronous process dependent on FSM state and FSM inputs
    - Use of if, then, else for scalar decision
    - Use of case for vector decisions
  - Illustration using binary multiplier ASM chart in Figure 8-8 of text

```
-- Alternate Binary Multiplier with n = 4
-- See Figure 8-8 of text
library ieee;
use ieee.std logic 1164.all;
use ieee.std logic unsigned.all;
entity alt bin multiplier is
port(CLK, RESET, G, LOADB, LOADQ: in std logic;
      MULT IN: in std logic vector(3 downto 0);
      MULT OUT: out std logic vector(7 downto 0));
end alt bin multiplier;
architecture sequential of alt bin multiplier is
type state type is (IDLE, MUL);
signal state, next state: state type;
signal P: std logic vector(1 downto 0);
signal A,B,Q: std logic vector(3 downto 0);
signal Z: std logic;
```

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

```
begin
-- Test P for value 0:
Z \le not(P(1) \text{ or } P(0));
-- state register
state register: process (CLK, RESET)
begin
 if (RESET = '1') then
  state <= IDLE;</pre>
 elsif (CLK'event and (CLK = '1')) then
  state <= next state;</pre>
 end if;
end process;
-- next state function
ns function: process (state, G, Z)
begin
 Logic and Computer Dealgn Fundamentals
 2804 Pearson Education, Inc.
```

VHDL - Part 5 4

```
if (state = IDLE) then
  if (G = '1') then
   next state <= MUL;</pre>
  else
   next state <= IDLE;</pre>
  end if;
 else -- state = MUL
  if (Z = '1') then
   next state <= IDLE;</pre>
  else
   next state <= MUL;</pre>
  end if;
 end if;
end process;
-- register transfers
reg trans: process (CLK)
variable AQ: std logic vector(7 downto 0);
 Logic and Computer Design Fundamentals
  PowerPoint<sup>®</sup> Slides
  2804 Pearson Education, Inc.
```

```
begin
if (CLK'event and (CLK = '1')) then
 if (LOADB = '1') then
  B \leq MULT IN;
 elsif (LOADO = '1') then
  Q \leq MULT IN;
 elsif ((state = IDLE) and (G = '1')) then
    P <= "11";
    A <= "0000";
 elsif (state = MUL) then
     P \le P - "01";
     if (O(0) = '1') then
      AQ := (('0' \& A) + ('0' \& B)) \& Q(3 \text{ downto } 1);
-- The 0s preceding A and B permit the carry out of the
  addition to be captured instead of having an overflow.
```

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

```
else
       AQ := '0' & A & Q(3 downto 1); -- Zero fill is used
                  -- since this is an unsigned right shift.
     end if;
     A \leq AQ(7 \text{ downto } 4);
     Q \leq AQ(3 \text{ downto } 0);
    end if;
end if;
end process;
MULT OUT \leq A & Q;
end sequential;
```

# **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