Teaching Sequential Logic VHDL Models by Synthesis and Examples

Guoping Wang

Department of Engineering, Indiana University Purdue University Fort Wayne

Abstract

VHDL has become an industrial standard language in digital system design. This paper introduces the author's experience in teaching sequential logic VHDL models to students through synthesis and examples from simple to complex design problems. The simple sequential circuits such as latches, FFs (flip-flops) are first introduced with all the control signals, then the same design concepts and procedures are extended to sequential logic blocks such as counters and shift registers. These design approaches are also applicable in complex sequential digital system designs. The author's experiences showed the effectiveness of this approach in teaching sequential logic VHDL models.

Index term: Engineering course, VHDL, digital systems, sequential circuit.

Introduction

The VHSIC (very high speed integrated circuits) Hardware Description Language (VHDL) is a very powerful hardware language for digital system design. It has become indispensable in electrical and computer engineering programs.

Digital logic systems can be classified in combinational and sequential logic circuits. Sequential logic systems play a very important part. In teaching VHDL models of sequential logic circuits, the author found that students could easily get confused with all the different control signals, such as, CLOCK, ENABLE, RESET/SET, OVERFLOW, CARRY-OUT etc. Various VHDL teaching methods have been proposed in the past [1-7], however, how to teach sequential logic VHDL models more effectively has not been studied so far.

This paper introduces the author's experience in teaching sequential logic VHDL models to students through synthesis and examples from simple to complex design problems. The simple sequential circuits such as latches, flip-flops are first introduced with all the control signals, then the same design concepts and procedures are extended to the designs of sequential logic blocks such as counters and shift registers. These design approaches are also applicable in complex sequential digital system designs. The author's experiences showed the effectiveness of this approach in teaching sequential logic VHDL models. During the teaching process, both of the VHDL models and the synthesized circuits are introduced. This teaching approach is design-

oriented instead of focusing too much on the complexities of VHDL language, such as package, library and different delays.

Selection of Textbooks

Most VHDL textbooks use models developed for simulation only and they are armed at practising engineers. They frequently use language features not supported in synthesized circuit and they are not easy for beginners to read [8-9]. The textbook *Digital System Design with VHDL* by M. Zwolinski [10] was chosen because it mainly focuses upon the design circuit using VHDL instead of the complex features of this language. Although some VHDL features are not covered, this book is a good textbook for university students. Not only lots of VHDL example codes are introduced in this book, the synthesized circuits are also provided.

Basic Sequential Logic Block VHDL Model

VHDL models of D flip-flops are explained using the features of asynchronous /synchronous RESET/SET, clock enable, output enable, different signal outputs and different output styles. The corresponding logic diagrams are synthesized to compare the VHDL codes and the circuits. For example, an asynchronous RESET/SET D flip-flop is synthesized from VHDL model in Figure 1, while a synchronous RESET/SET D flip-flop is synthesized from VHDL model in Figure 2. Notice that in Figure 1, the conditional signal assignments of "QQ:='0', QQ:='1'" appear before *CLK'event and CLK='1'*, while in Figure 2, they appear inside the *CLK'event and CLK='1'* statement.

```
 if (RESET='1') then
    QQ:='0';
elsif (SET='1') then
     QQ:='1';
elsif (CLK'event and CLK='1') then
 …….
```

Figure 1. An Asynchronous RESET/SET D Flip-Flop VHDL Model

```
if (CLK'event and CLK='1') then
      if (SET='1') then
         QQ:='1';
    elsif (RESET='1') then
         QQ:='0';
     else
…….
```

Figure 2. A Synchronous RESET/SET VHDL Model

A D flip-flop with two-register Q, Q-complement outputs is synthesized from VHDL model in Figure 3, while a D flip-flop with one register Q, Q-complement outputs is synthesized from VHDL model in Figure 4. Notice that in Figure 3, both Q, Qbar assignments appear inside the *Clk'event and CLK='1'*, while in Figure 4, only "QQ:=D" appears inside the *Clk'event and CLK='1'* statement.

```
if (CLK'event and CLK='1') then
      Q<=D;
```

```
            Qbar<=not D;
end if;
……
```

Figure 3. D FF VHDL Model with Synthesized Two Registers

```
if (CLK'event and CLK='1') then
        QQ:=D;
end if;
Q<=QQ;
QBAR<= not QQ;
```

Figure 4. D FF VHDL Model with Synthesized One Register

The VHDL models with clock enable and tri-state output are presented in Figures 5 and 6.

```
if (CLK'event and CLK='1') then
   if (EN='1') then
         QQ:=D;
  end if;
```

Figure 5. DFF VHDL Model with Clock Enable

```
If CLK'event and CLK='1' then
…..
end if;
if (OE='1') then
        Q<='Z'; QBAR<='Z';
else
        Q<=QQ; QBAR<=not QQ;
end if;
```

Figure 6. DFF VHDL Model with Tri-state Output Enable

After VHDL models are introduced, the synthesized logic circuits are also explained to the students to help them better understand the concepts of these different control signals. For example, the logic diagram for D FF with synthesized two registers is shown in Figure 7. A D FF with output enable and tri-state output is shown in Figure 8.

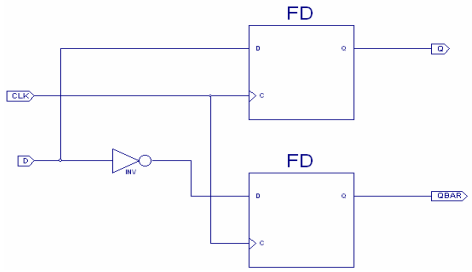

Figure 7. D FF with Synthesized Two Registers

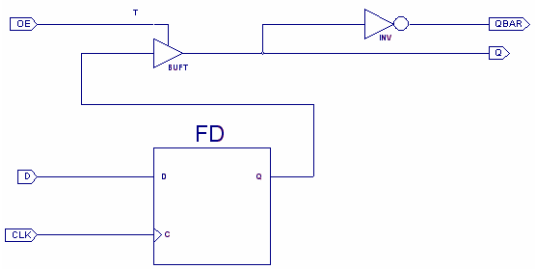

Figure 8. D FF with Output Enable and Tri-state Output

After that, an example is introduced to give the students a complete picture of the D flip-flop with all these control signals such as, asynchronous reset/set, clock enable and output enable. Figure 9 shows such a VHDL model, while Figure 10 shows the corresponding logic circuit.

```
entity DFF_COMBINE is
   Port ( D :      in   std_logic;
          CLK :    in   std_logic;
          SET :    in   std_logic;
          RESET : in   std_logic;
          EN :     in   std_logic;
          OE :     in   std_logic;
          Q :      out std_logic;
          QBAR :   out std_logic);
end DFF_COMBINE;
architecture Behavioral of DFF_COMBINE is
begin
process(D,CLK,SET,RESET,EN,OE)
          variable QQ:std_logic;
          begin
          if (RESET='1') then
                QQ:='0';
          elsif (SET='1') then
                QQ:='1';
          elsif  (CLK'event and CLK='1') then
                  if (EN='1') then
                          QQ:=D;
                    end if;
            end if;
            if (OE='0') then Q<=QQ; else Q<='Z'; end if;
            if (OE='0') then Qbar<=not QQ; else Qbar<='Z'; end if;
     end process;
end Behavioral;
```

Figure 9. A Complete D Flip-Flop VHDL Model

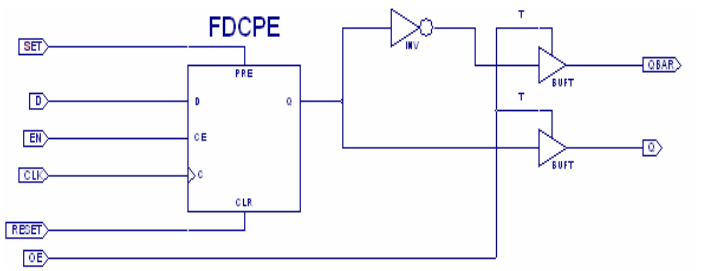

Figure 10. D Flip-Flop with Asynchronous Reset/Set, Clock Enable, Output Enable

Sequential Logic Block Design

After the simple D FFs are introduced, the VHDL models for moderately complex sequential logic blocks are introduced to the students. The design procedures are illustrated through two design examples in this paper as the following.

1) Design a VHDL model for the 74x173. The truth table is given in the following.

TRUTH TABLE

INPUTS				DATA	OUTPUT
		DATA ENABLE			
MR	CP	$\overline{E1}$	$\overline{E2}$	D	Q_n
H	X	X	X	X	L
L	L	X	X	X	Q_0
L	↑	H	X	X	Q_0
L	↑	X	H	X	Q_0
L	↑	L	L	L	L
L	↑	L	L	H	H

H= High Voltage Level
L = Low Voltage Level
X= Irrelevant
↑= Transition from Low to High Level
Q_0= Level Before the Indicated Steady-State Input Conditions Were

From the truth table, 74173 has an asynchronous RESET signal – MR, two CLOCK enable signals, E1 and E2, and it is a rising-edge triggered D FF. Thus, the VHDL code can be written as Figure 11:

```
entity IC74173 is
   Port ( D : in std_logic;
          MR : in std_logic;
          CP : in std_logic;
          E1 : in std_logic;
          E2 : in std_logic;
          Q : out std_logic);
          end IC74173;
```

```
architecture behavioral of IC74173 is
begin
        process(D,MR,CP,E1,E2)
          begin
                if MR='1' then
                  Q<='0';
              elsif rising_edge(CP) then
                    if (E1='0') and (E2='0') then
                        Q<=D;
                    end if;
                end process;
end behavioral;
```

Asynchronous RESET

Two clock enable signals

Figure 11. VHDL Model for 74x173

2) VHDL model for 74x190 counter.

The 74x190 is asynchronously presettable BCD decimal counters. Presetting the counter to the number on preset data inputs (A−D) is accomplished through a low asynchronous parallel load (LOAD) input. Counting occurs when LOAD is high, count enable (CTEN) is low, and the down/up (D/U) input is either high for down counting or low for up counting. The counter is decremented or incremented synchronously with the low-to-high transition of the clock.

When an overflow or underflow of the counter occurs, the MAX/MIN output, which is low during counting, goes high and remains high for one clock cycle. This output can be used for look-ahead carry in high-speed cascading. The MAX/MIN output also initiates the ripple clock (RCO) output, which normally is high, goes low, and remains low for the low-level portion of the clock pulse. These counters can be cascaded using RCO. If a decade counter is preset to an illegal state or assumes an illegal state when power is applied, it returns to the normal sequence in one or two counts, as shown in the state diagrams.

FUNCTION TABLE

INPUTS				FUNCTION
$\overline{LOAD}$	$\overline{CTEN}$	D/$\overline{U}$	CLK	
H	L	L	⌐	Count up
H	L	H	⌐	Count down
L	X	X	X	Asynchronous preset
H	H	X	X	No change

D/$\overline{U}$ or $\overline{CTEN}$ should be changed only when clock is high.
X = Don't care
⌐ Low-to-high clock transition

From the functional description, LOAD is an asynchronous PRESET signal, CTEN is a Clock enable signal, Maxmin and RCO are the outputs which only depend upon the counter states. Figure 12 shows the schematic design diagram.

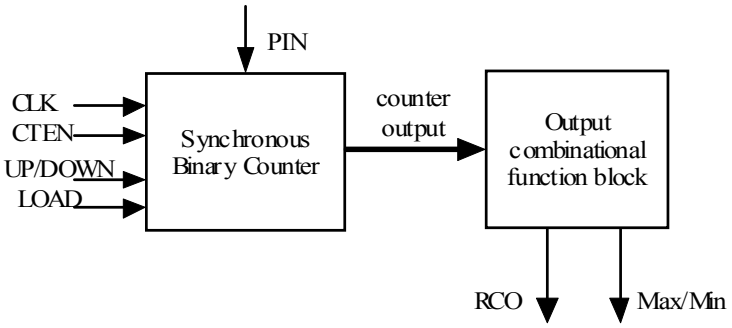

Figure 12. 74x190 Schematic Model

The VHDL mode for 74x190 is written as Figure 13.

```
entity IC74LS190 is
   Port (
        RST:          in std_logic;
        PIN :         in std_logic_vector(3 downto 0);
        CTEN :        in std_logic;
        LOAD :        in std_logic;
        DOWN_UP: in std_logic;
        CLK :         in std_logic;
        Q :           out std_logic_vector(3 downto 0);
        RCO :         out std_logic;
        MAX_MIN: out std_logic);
end IC74LS190;

architecture Behavioral of IC74LS190 is
   signal QQ:unsigned(3 downto 0);
begin
  count_pro:
  process(RST,PIN,CTEN,LOAD,DOWN_UP,CLK,QQ)
  begin
    if RST='1' then                                  Asynchronous Reset and Load
         QQ<=(others=>'0');
      elsif LOAD='0' then
          QQ<=unsigned(PIN);                         Clock enable
      elsif (CLK'event and CLK='1') then
           if (CTEN='0') then
              if DOWN_UP='1' then
                    if QQ="1001" then
                       QQ<="0000";
                    else
                       QQ<=QQ+1;
              end if;
            else
                  if QQ="0000" then
                      QQ<="1001";
                  else
                      QQ<=QQ-1;
                  end if;
```

```
            end if;
          end if;
        end if;
end process;
 Q<=std_logic_vector(QQ);

rco_max_min_pro: process(CLK,QQ,CTEN,DOWN_UP)
     begin
       RCO<='1';
       if (CLK='0' and CTEN='0' and DOWN_UP='0' and QQ="0000") then
              RCO<='0';
      elsif (CLK='0' and CTEN='0' and DOWN_UP='1' and QQ="1001") then
              RCO<='0';
       end if;
      MAX_MIN<='0';
      if (CTEN='0' and DOWN_UP='0' and QQ="0000") then
           MAX_MIN<='1';
      elsif (CTEN='0' and DOWN_UP='1' and QQ="1001") then
           MAX_MIN<='1';
     end if;
 end process;
end behavioral;
```

Output enable signals and output combinational logic

Figure 13. VHDL Model for 74x190

Students Feedback

The introduction of simple D FFs prior to the sequential logic blocks gave students an easy start on the way to sequential logic designs using VHDL. During the lecture and lab sessions, students showed very positive feedback. In the exam and lab projects, most students demonstrated the ability to design moderately complex sequential logic blocks. The course and the lab sessions were rated 3.69/4.0 overall in the 2004 Spring semester.

Conclusion

VHDL is a challenging course for electrical and computer engineering students due to its complex features. This paper introduces a new approach to teaching sequential VHDL models. The simple D flip-flop VHDL models are introduced first, and then the design concepts are migrated to the complex logic blocks. The author's experience in IPFW (Indiana University Purdue University Fort Wayne) shows that it is very effective in helping students focus on the design and implementation of VHDL instead of the complex language features.

References

[1] A. Etxebarria, I. J. Oleagordia, and M. Sanchez, "An educational environment for VHDL hardware description language using the WWW and specific workbench," in *Proceedings - Frontiers in Education Conference*, 2001, vol. 1, pp. 22-27.

[2] A. I. Hussein, D. M. Gruenbacher, and N. M. Ibrahim, "Design and verification techniques used in a graduate level VHDL course," *Proceedings - Frontiers in Education Conference*, vol. 2 pp. 13-14, 1999.
[3] P. L. Jones, "Getting started with VHDL," in *Proceedings of the IEEE International Conference on Microelectronic Systems Education, MSE*, 1997, pp. 135-136.
[4] A. Wu, "Interactive learning toolbox for logic synthesis with VHDL," in *Proceedings of the IEEE International Conference on Microelectronic Systems Education, MSE*, 1997, pp. 77-78.
[5] M. Chang, "Teaching top-down design using VHDL and CPLD," in *Proceedings - Frontiers in Education Conference*, 1996, vol. 2, pp. 514-517.
[6] J. Pedraza-Chavez, D. Baez-Lopez, and J. M. Ramirez, "Use of VHDL as a tool for the teaching of digital systems," in *ASEE Annual Conference Proceedings*, 1995, vol. 2, pp. 2800-2802.
[7] V. Pedroni, "Teaching Design-Oriented VHDL", in *Proceedings of the 2003 IEEE International Conference on Microelectronic Systems Education, MSE'03,* pp.
[8] Z. Navabi, *VHDL Analysis and Modeling of Digital Systems*, McGraw-Hill, 1998.
[9] D. L. Perry, *VHDL,* New York: McGraw-Hill, 1994.
[10] M. Zwolinski, *Digital System Design with VHDL*, Prentice Hall, 2000.

BIOGRAPHICAL INFORMATION

GUOPING WANG is currently Assistant Professor in the Department of Engineering, Indiana University Purdue University Fort Wayne. He obtained his B.S., M.S., and Ph.D from Tsinghua University, Nanjing University and the University of Oklahoma respectively in 1988, 1991 and 2003. He teaches courses in digital system design, VLSI Design Lab, and computer architecture.