AC 2007-1289: THE INNOVATIVE EFFECTS OF HDL AND FPGA ON DIGITAL HARDWARE DESIGN EDUCATION IN EET PROGRAMS

Hong "Jeffrey" Nie, University of Northern Iowa

Hong (Jeffrey) Nie is an assistant professor of Electrical and Information Engineering Technology Program at UNI. Dr. Nie received his Ph.D. in EE from the University of British Columbia, Canada in 2003. His research interests are in the area of hardware implementations of wireless sensor networks, advanced wireless transceiver architectures for sensors in harsh electromagnetic, ultra wide-band techniques for wireless sensor transceivers, and software defined radio & related digital signal processing techniques. Dr. Nie’s recent research projects include wireless sensor and digital signal processing technologies for data acquisition systems in machine condition monitoring, and ultra wide-band techniques for wireless sensor transceivers in industrial environments. Before Dr. Nie joined UNI in Aug. 2006, he won several federal level research grants in Canada as a principle investigator.

Recayi "Reg" Pecen, University of Northern Iowa

RECAYI “Reg” PECEN Dr. Pecen holds a B.S.E.E. and an M.S. in Controls and Computer Engineering from the Istanbul Technical University, an M.S.E.E. from the University of Colorado at Boulder, and a Ph.D. in Electrical Engineering from the University of Wyoming (UW). He has served as faculty at the UW, and South Dakota State University. He is currently an associate professor and program coordinator of Electrical and Information Engineering Technology program in the Department of Industrial Technology at the University of Northern Iowa. His research interests and publications are in the areas of AC/DC Power System Interactions, power quality, and grid-connected renewable energy applications. He is a member of ASEE, IEEE, Tau Beta Pi National Honor Society, and NAIT. Dr. Pecen was recognized as an Honored Teacher/Researcher in “Who’s Who among America’s Teachers” in 2004 and 2005. He was also nominated for 2004 UNI Book and Supply Outstanding Teaching Award, March 2004, and nominated for 2007 Russ Nielson Service Award, UNI. He serves as a Center for Energy Environment Education (CEEE) Research fellow at UNI in 2007.
The Innovative Effects of Hardware Description Languages and Field Programmable Gate Arrays on Digital Hardware Design Education in Engineering Technology Programs

Abstract

With the development of computer-aided design and semiconductor technologies, Hardware Description Language (HDL) and Field Programmable Gate Arrays (FPGA) have become dominant design and implementation tools for digital hardware and are widely used by both industry and academia. As a result, they will bring many innovative effects on the architectures, contents, and methodologies of digital hardware design courses in Electrical Engineering Technology (EET) programs. In this paper, by comparing in detail the design procedures of three typical digital circuits when the traditional design methodology and the VHDL and FPGA methodology are employed respectively, we will clearly demonstrate those innovative effects of VHDL and FPGA on digital hardware design courses, such as why some traditional skills are outdated and what the newly emerged skills are. From the design procedures of those typical digital circuits, we can conclude that the best way to teach HDL and FPGA in EET programs is to integrate HDL and FPGA into all digital hardware design courses, from entry level to advanced level, rather than open an independent course.

I. Introduction

Digital hardware design has been considered as core education contents for Electrical Engineering Technology (EET) programs for more than two decades. As a result, a whole set of methodologies, such as truth table, canonical sum-of-products expressions, and Karnaugh maps for combinational circuit, and finite state machine, state diagram, state table, and state assignment for sequential circuit, have been well-developed to analyze and design digital circuits. However, because the complexity to design a digital circuit increases exponentially with the number of gates used in the circuit, only small-scale digital circuits can be explored in digital hardware design courses when those traditional methodologies are employed.

In order to analyze and design large-scale digital circuits, Hardware Description Languages (HDLs), including Verilog and VHDL (Very High Speed Integrated Circuits HDL), have been developed to describe the model and behavior of digital hardware. Although HDLs are similar to a computer programming language in format, HDL codes are not programs to be executed on a computer. HDLs were originally developed for two purposes: first, as a documentation language to describe the structure and behavior of complex digital circuits designed by multiple designers; second, as an input to computer simulation software to simulate the operation of circuits. Since VHDL was first established as the IEEE 1076 standard in 1987, and then revised as the IEEE 1164 standard in 1993, many Computer-Aided Design (CAD) systems adopt VHDL to provide documentation and simulation functions. Furthermore, with the development of CAD technologies, more and more CAD system also use VHDL as design entry to provide synthesis functions, i.e. converting VHDL codes into a hardware implementation of the described circuit. Now VHDL has become a dominant hardware developing tool to design, simulate and document large-scale digital circuits and is widely used by both industry and academia.
Meanwhile, as a result of fast growth of semiconductor industry, Field Programmable Gate Arrays (FPGAs)⁵, which are large-scale Integrated Circuit (IC) chips containing a large number of programmable logic gates and programmable internal interconnects, become more and more economically available. For example, as shown in Fig. 1, Digilent Inc. provides a FPGA development board, which consists of a Xilinx Spartan-3 FPGA with 200,000 gates, a fast asynchronous SRAM with 1Mega bytes, and various on-board I/O devices, with $99 per board⁶. As compared to breadboards and small-scale digital IC chips (e.g. 74xx and 4000 series) commonly used in traditional digital circuit design courses, FPGA development boards can be used to implement much more complex digital functions with much less external interconnects. However, the employment of FPGA development boards is strongly connected with HDLs. As shown in Fig. 2, first, VHDL codes are written to describe the model and behavior of the desired digital circuit; second, those VHDL codes are synthesized and converted to a programming file
by CAD systems; finally, the programming file is downloaded to the FPGA development board to program the logic gates and internal connections of the FPGA to implement the desired digital circuit. Clearly, once VHDL and the FPGA development board are combined together and employed in digital hardware design courses, they will create many innovative effects on the architectures, contents, and methodologies of digital hardware design education.

To the best of our knowledge, most electrical engineering programs in US universities have integrated VHDL and FPGA into their digital hardware design courses. However, EET programs in US universities are relatively slow in moving from the traditional digital hardware design education to a VHDL and FPGA integrated one. By checking through Internet the course descriptions for digital hardware design courses, we have found that among 60 US universities and colleges offering EET or similar programs, only six universities (Arizona State University, DeVry University, Ferris State University, Purdue University, Saint Louis University, and University of Hartford) have integrated VHDL and FPGA into their digital hardware design courses. Another six universities and colleges offer an independent course for VHDL and FPGA in their EET programs. It seems that the rest 48 universities and colleges still teach digital hardware design courses in the traditional way.

Some educators in EET programs have noticed that VHDL and FPGA will bring innovative effects on digital hardware design education. For example, when new technologies like VHDL and FPGA are employed in digital circuit design, newly emerged skills and outdated skills are listed in reference. In this paper, in order to clearly demonstrate the innovative effects of VHDL and FPGA on digital hardware design courses, i.e. why some traditional skills are outdated and what the newly emerged skills are, we will compare in detail the design procedures of three typical digital circuits when the traditional design methodology and the VHDL and FPGA methodology are employed respectively. The organization of this paper is as follows. After this introduction, the design cases of a ballot counter counting the number of ballots supporting or against a motion, which is a typical combinational logic circuit, and a clock counter with specific output waves, which is a typical sequential logic circuit, are studied respectively in Section II and III. Then in Section IV, a complex digital circuit design project is considered: a four-digit frequency meter with three frequency scales (up to 10kHz, up to 100kHz, and up to 1,000kHz). Finally, the effective ways to integrate VHDL and FPGA into digital hardware design courses in EET programs are discussed, and conclusions are given in Section V.

II. Design of a Ballot Counter

The ballot counter to be design has five inputs and three outputs. The number of ‘1’ of the five inputs should be indicated at the three outputs in a binary format. According to the traditional digital circuit design methodology, the design procedures consist of the following three steps.

First, a truth table shown in Table 1 is created. Second, based upon this truth table, the canonical sum-of-products expressions for $y_2$, $y_1$, and $y_0$ are derive as follows:

$$y_2 = \sum m(15,23,27,29,30,31) ,$$

$$y_1 = \sum m(3,5,6,7,9,10,11,12,13,14,17,18,19,20,21,22,24,25,26,28) ,$$

$$y_0 = \sum m(1,2,4,7,8,11,13,14,16,19,21,22,25,26,28,31) .$$
Then for each of the canonical sum-of-products expressions, a five-variable Karnaugh map is used to reduce the number of product or sum terms in the expression so as to obtain a minimum-cost expression that uses the minimum number of gates and interconnections. Finally, according to the obtained minimum-cost expressions, the ballot counter is implemented by connecting small-scale IC chips with wires on a breadboard. Obviously, the above-described design procedures are long and tedious. Students are easy to make mistakes during the design and implementation procedures, especially when they deal with those five-variable Karnaugh maps.

On the contrary, when the VHDL and FPGA methodology is employed to design this ballot counter, the design procedures are easy and straightforward. First, the inputs and outputs of the ballot counter are defined with a VHDL construct called entity as follows:

\[
\text{entity ballot is}
\text{Port ( X : in \STD_LOGIC\_VECTOR (4 downto 0);}
\text{ Y : out \STD_LOGIC\_VECTOR (2 downto 0));}
\text{end ballot;}
\]

Here we have assigned a name to the designed entity as ballot. Second, the functionality of the ballot counter is specified with another VHDL construct called architecture as follows:

\[
\text{architecture ballotfunc of ballot is}
\text{begin}
\text{with X select}
\text{Y <= "000" when "00000",}
\text{"001" when "00001"|"00010"|"00100"|"01000"|"10000",}
\text{"010" when "00011"|"00101"|"01001"|"10001"|"00110"|"01010"|}
\text{"10010"|"01100"|"10100"|"11000",}
\text{end ballot;}
\]

Table 1. Truth Table for Ballot Counter

<table>
<thead>
<tr>
<th>x4</th>
<th>x3</th>
<th>x2</th>
<th>x1</th>
<th>x0</th>
<th>y2</th>
<th>y1</th>
<th>y0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0</td>
<td>0 0</td>
<td>0 0</td>
<td>0 0</td>
<td>0 0</td>
<td>0 1</td>
<td>0 0</td>
<td>0 0</td>
</tr>
<tr>
<td>0 0</td>
<td>0 0</td>
<td>0 0</td>
<td>0 1</td>
<td>0 0</td>
<td>0 1</td>
<td>0 0</td>
<td>0 1</td>
</tr>
<tr>
<td>0 0</td>
<td>0 0</td>
<td>1 0</td>
<td>0 0</td>
<td>1 0</td>
<td>0 0</td>
<td>0 1</td>
<td>0 0</td>
</tr>
<tr>
<td>0 0</td>
<td>0 0</td>
<td>0 0</td>
<td>1 1</td>
<td>0 0</td>
<td>0 1</td>
<td>0 1</td>
<td>0 1</td>
</tr>
<tr>
<td>0 0</td>
<td>0 1</td>
<td>1 0</td>
<td>0 1</td>
<td>1 0</td>
<td>1 0</td>
<td>0 1</td>
<td>0 1</td>
</tr>
<tr>
<td>0 0</td>
<td>0 0</td>
<td>0 0</td>
<td>0 1</td>
<td>0 0</td>
<td>0 1</td>
<td>0 0</td>
<td>0 1</td>
</tr>
<tr>
<td>0 0</td>
<td>0 1</td>
<td>1 0</td>
<td>0 1</td>
<td>1 0</td>
<td>1 0</td>
<td>0 1</td>
<td>0 1</td>
</tr>
<tr>
<td>0 0</td>
<td>0 1</td>
<td>1 0</td>
<td>0 1</td>
<td>1 0</td>
<td>1 0</td>
<td>0 1</td>
<td>0 1</td>
</tr>
<tr>
<td>0 0</td>
<td>0 1</td>
<td>0 0</td>
<td>0 1</td>
<td>0 0</td>
<td>1 0</td>
<td>0 1</td>
<td>0 1</td>
</tr>
<tr>
<td>0 0</td>
<td>0 1</td>
<td>0 0</td>
<td>0 1</td>
<td>0 0</td>
<td>1 0</td>
<td>0 1</td>
<td>0 1</td>
</tr>
<tr>
<td>0 1</td>
<td>0 1</td>
<td>1 0</td>
<td>0 1</td>
<td>0 1</td>
<td>1 0</td>
<td>0 1</td>
<td>0 1</td>
</tr>
<tr>
<td>0 1</td>
<td>0 1</td>
<td>1 0</td>
<td>0 1</td>
<td>0 1</td>
<td>1 0</td>
<td>0 1</td>
<td>0 1</td>
</tr>
<tr>
<td>0 1</td>
<td>1 1</td>
<td>0 0</td>
<td>0 1</td>
<td>0 1</td>
<td>1 0</td>
<td>0 1</td>
<td>0 1</td>
</tr>
<tr>
<td>0 1</td>
<td>1 1</td>
<td>0 0</td>
<td>0 1</td>
<td>0 1</td>
<td>1 0</td>
<td>0 1</td>
<td>0 1</td>
</tr>
<tr>
<td>0 1</td>
<td>1 1</td>
<td>1 0</td>
<td>0 1</td>
<td>1 1</td>
<td>1 0</td>
<td>1 0</td>
<td>0 0</td>
</tr>
<tr>
<td>0 1</td>
<td>1 1</td>
<td>1 0</td>
<td>0 1</td>
<td>1 1</td>
<td>1 0</td>
<td>1 0</td>
<td>0 0</td>
</tr>
<tr>
<td>0 1</td>
<td>1 1</td>
<td>1 0</td>
<td>0 1</td>
<td>1 1</td>
<td>1 0</td>
<td>1 0</td>
<td>0 0</td>
</tr>
<tr>
<td>0 1</td>
<td>1 1</td>
<td>1 0</td>
<td>0 1</td>
<td>1 1</td>
<td>1 0</td>
<td>1 0</td>
<td>0 0</td>
</tr>
</tbody>
</table>
Here we have assigned a name to the designed architecture as \textit{ballotfunc}. In this architecture, we simply specify the expecting values of $Y$ according to the input values of $X$. Since the FPGA development board shown in Fig. 1 has eight switches and eight LEDs, an accessory file is edited to connect $X$ to five switches and $Y$ to three LEDs. Finally, the synthesis functions of the CAD system are employed to create a programming file from the VHDL codes and the accessory file, and the obtained programming file is downloaded to the FPGA development board to implement the design. Clearly, when the VHDL and FPGA methodology is employed, we do not need to derive the canonical sum-of-products expressions given by Eqs. (1)-(3); we do not need to simplify the canonical sum-of-products expressions with the complicated five-variable Karnaugh maps; and we do not need to use small-scale IC chips, wires and a breadboard to implement the circuit. All those routine tasks are now implemented by the CAD systems. Consequently, with the VHDL and FPGA methodology, based upon our teaching experience, most EET student can design and implement a ballot counter within half hour, but with the traditional digital circuit design methodology, an instructor may not be able to implement the same circuit for two hours.

III. Design of a Clock Counter with Specific Output Waves

The clock counter under consideration has two inputs and four outputs. One input is a 50MHz clock, and the other input is a reset which forces all outputs to be ‘1’ when it is ‘1’. When the reset is ‘0’, the four outputs must have the specific waves shown in Fig. 3, and the duration of each output to be ‘0’ is 1ms. When the traditional digital circuit design methodology is employed, multiple sequential and combinational logic units must be combined together to achieve the required functionality. For example, the top-level block diagram for a possible realization is shown in Fig. 4. Many further design activities must be performed to implement the three logic units shown in Fig. 4 with small-scale IC chips, especially for the 5000-dividing counter.

![Fig. 3. Output Waves for the Clock Counter](image-url)
On the other hand, when the VHDL and FPGA methodology is employed, the clock counter can be designed with the VHDL codes as follows:

```vhdl
entity counter is
    Port ( CLK : in STD_LOGIC;
          RESET : in STD_LOGIC;
          AN : out STD_LOGIC_VECTOR (3 downto 0));
end counter;

architecture Counterfunc of counter is
begin
    process (CLK, RESET) is
    variable refresh : std_logic_vector(17 downto 0);
    begin
        if RESET='1' then
            AN <= "1111";
            refresh := (others => '0');
        elsif rising_edge(CLK) then
            refresh := refresh + 1;
            if refresh < 50000 then
                AN <= "1110";
            elsif refresh < 100000 then
                AN <= "1101";
            elsif refresh < 150000 then
                AN <= "1011";
            elsif refresh < 200000 then
                AN <= "0111";
            else
                AN <= "1110";
                refresh := (others => '0');
            end if;
        end if;
    end if;
end process;
end Counterfunc;
```

In this VHDL codes, again instead of describe how to realize the counter with digital logic, we simply specify the desired behaviors of the counter, which can be summarized as follows:

1. If \( \text{RESET} = '1' \), \( \text{AN} = "1111" \) and \( \text{refresh} = 0 \);
(2) For every rise edge of \( CLK \), \( \text{refresh} \) adds one;
(3) When \( 0 \leq \text{refresh} < 5000 \), \( AN=“1110” \), \( 5000 \leq \text{refresh} < 10000 \), \( AN=“1101” \),
\( 10000 \leq \text{refresh} < 15000 \), \( AN=“1011” \), and \( 15000 \leq \text{refresh} < 20000 \), \( AN=“0111” \);
(4) If \( \text{refresh} \) is other value, let \( \text{refresh}=0 \), and \( AN=“1110” \).

Then the rest design tasks are simply and straightforward: editing an accessory file to connect \( AN, \text{CLK}, \text{RESET} \) to the suitable ports in the FPGA development board, creating a programming file with the synthesis functions of the CAD systems, and downloading to the FPGA development board.

IV. Design of a Three-Scale Four-Digit Frequency Meter

The desired frequency meter should have three scales: up to 10kHz, up to 100kHz and up to 1MHz. The switching between the three scales is determined by two ‘0’/’1’ switches. The results obtained by the frequency meter should be shown on a 4-digital, seven-segment LED as follows:

(1) For the 10kHz scale: If the frequency of the input signal is lower than 10kHz, the LED shows “5.555” for 5.555kHz, “0.999” for 999Hz, “0.034” for 34Hz, or “0.005” for 5Hz. Otherwise, the LED should be shown as “E.EEE”.
(2) For the 100kHz scale: If the frequency of the input signal is lower than 100kHz, the LED shows “55.55” for 55.55kHz, “09.99” for 9.99kHz, “00.34” for 340Hz, or “00.05” for 50Hz. Otherwise, the LED should be shown as “EE.EE”.
(3) For the 1MHz scale: If the frequency of the input signal is lower than 1MHz, the LED shows “555.5” for 555.5kHz, “099.9” for 99.9kHz, “003.4” for 3.4kHz, or “000.5” for 500Hz. Otherwise, the LED should be shown as “EEE.E”.

Furthermore, it is required that for this frequency meter the longest refresh time between each display cannot be longer than one second.

Fig. 5. A Three-Scale Four-Digit Frequency Meter Realized with the FPGA Development Kit

If this frequency meter is designed and realized with the traditional digital design methodology, even in the point of view of an instructor, it would be a quite complex digital project, and EET
students would need to have a dedicated course like Senior Design to implement this project. However, if the VHDL and FPGA methodology is employed, the frequency meter can be designed with VHDL codes less than 300 lines, and as shown in Fig. 5, the frequency meter implemented by the FPGA development board can precisely measure the frequency of the signal outputted by a signal generator. Based upon our teaching experience, if this project is used as an in-course project for the second digital circuit design course in EET program, most students can implement the frequency meter with the VHDL and FPGA methodology.

V. Discussions and Conclusions

From the three case studies demonstrated in Sections II, III, and IV, it is clear that with more and more CAD system provide synthesis functions, i.e. automatically converting HDL codes into a hardware implementation of the described circuit, the design emphasis for digital circuits has changed from how to realize the digital logic to how to describe the model and behavior of the digital logic. As a result, many traditional design methodologies, for example, Karnaugh map for combination logic, and state assignment for sequential logic, which are fundamental digital circuit design skills in the past, become outdated when HDL and FPGA are employed in digital hardware design, since those traditional design methodologies are mainly used to realize the digital logic instead of describing the model and behavior of the digital logic.

It is well known that in traditional digital circuit design courses, so much time and efforts have been spent to train students to use Karnaugh map and state assignment, because without the help of synthesis functions in the CAD systems, Karnaugh map and state assignment are prerequisite methods to implement digital circuit design. Therefore, we can conclude that the best way to teach HDL and FPGA in EET programs is to integrate HDL and FPGA into all digital hardware design courses, from entry level to advanced level, rather than open a new course. By integrating HDL and FPGA into all digital hardware design courses, we can transfer the time and efforts previously spent in studying Karnaugh map and state assignment to study HDL and FPGA, and hence the total load of the digital circuit design courses would not increased. Furthermore, with the employment of HDL and FPGA, students may have more opportunities to explore the design of large-scale digital circuits with the same digital hardware design methodologies currently used in industry.

In summary, integrating HDL and FPGA into digital circuit design courses will bring many innovative effects on digital hardware design education in EET programs. Meanwhile, with the development of semiconductor and CAD technologies, not only the FPGA development kits, but also the CAD systems supporting VHDL synthesis become economically available. For example, through Xilinx’s University program, any university can apply for a donation from Xilinx to get its CAD systems for digital circuit design for free. So it is right time for EET programs in US universities to move from the traditional digital hardware design education to a VHDL and FPGA integrated one.

References

[6] [http://www.digilentinc.com](http://www.digilentinc.com)
[9] [http://www.xilinx.com/univ/don_program.htm](http://www.xilinx.com/univ/don_program.htm)