

Logic Design Using VHDL

Massoud Moussavi
Johnson & Wales University, School of Technology
Providence, R.I. 02903

Abstract: *In the digital system design world, VHDL (Very High Speed Integrated Circuit Hardware Description Language) has become an essential tool for designers. Understanding VHDL will help students successfully find a well paying position in today's job market. Nevertheless, not all engineering schools, especially those without a graduate program, offer a VHDL course. This paper intends to show how to integrate basic VHDL in the junior level Logic Design Course.*

Introduction:

VHDL is a design tool that contains text editors, compilers, debuggers, and simulators. Within the last decade, it has become a powerful tool for engineers to design fully tested and reliable digital systems. There are two major Hardware Descriptive Languages in the current market, AHDL and VHDL. AHDL is Altera's version of VHDL. Since Altera has a very supportive policy towards universities and colleges, educators may consider Altera's MAX + PLUS II in their digital circuit design class. Beside digital circuit design, VHDL has also been used in digital filter design.

Obviously, VHDL needs to be considered as a separate course in electronic/computer engineering. But adding a new course to the current curriculum comes with dropping one existing course and the approval of many time consuming committees. In order to shorten such a long process of approval, it is possible to teach basic of VHDL by integrating into a digital circuit design or a similar course. To fully understand VHDL, students may sign up for one of the available VHDL training courses, such as Aldec (www.aldec.com or info@aldec.com) or register for webtraining programs, such as Doulos' (www.doulos.co.uk). There are also many good books on VHDL that are listed at the end of this article.

To integrate VHDL in a logic design course, we may spend 15-20 percent of a typical 45 hour (total of 8-10 hours) class and teach basic VHDL through some examples. The following examples may provide a guideline for such an attempt.

Example 1. Entity and Architecture

The simple SOP combinational circuit of figure (1) is a good starting point to learn VHDL. In this example three basic gates: AND, Invertor, and OR gates (AOI) are connected together. The minterm expression of the output X is:

$$X(A,B,C) = (AB)' + (BC)$$

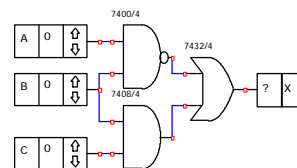


Figure 1. A simple combinational circuit.

A VHDL code typically starts with a definition and is followed by "entity" and "architecture". Thus, the VHDL model of figure (1) is as follow:

```

Line 1.      Library IEEE;
Line 2.      Use IEEE.STD_LOGIC_1164.all;
Line 3.      entity SOP is
Line 4. port
              (A,B,C: in    bit;
               X:      out  bit);
Line 5. end SOP;
Line 6. architecture structure of SOP is
              begin
                  X <= ((not A and B) or ( C and D));
Line 7.      end structure;

```

In this example, "SOP" and "structure" are the chosen names for the digital circuit. Words "in", "out", and "bit" show both the inputs and outputs are represented by one bit.

Example 2. Signal and Propagation Delay

In a digital circuit, it always takes a certain period of time for the output of each gate to respond to the events at its inputs. In a VHDL program, we should indicate the amount of propagation delay for an event on each signal. In the above example, input events on signals A,B, and C produce events on the internal signals (AB) and (BC). Events on signal (AB) produce events on (AB)'. Finally, events on internal signals (AB)' and (BC) will produce events on the output signal X. Thus, there are three signals (AB), (AB)', and (BC) in our digital circuit as shown in figure (2).

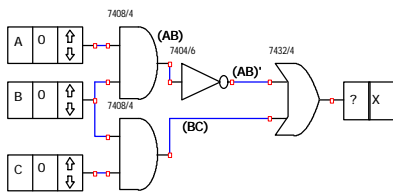


Figure 2. A combinational circuit with internal signals.

A change at inputs will transfer through these three signals and reach the output with a total propagation delay of 5ns (2ns for each of the 2-input gates and 1ns for the 1-input gate). Consequently, the above VHDL code will change as follows:

```

Line 1. Library IEEE;
Line 2. Use IEEE.STD_LOGIC_1164.all;
Line 3. entity SOP is
Line 4. port
      (A,B,C: in bit;
       X: out bit);
Line 5. end SOP;
Line 6. architecture structure of SOP is
      signal AB, AB', BC: std_logic:= '0';
      begin
          AB <= A and B after 2ns;
          BC <= B and C after 2ns;
          AB' <= not AB after 1ns;
          X <= AB' or BC after 2ns;
Line 7. end structure;
    
```

The above form of the signal assignment statement implicitly places a "after 0ns" time expression following the value expression.

Example 3. Components

A digital circuit is a interconnection of many components. The VHDL code describes the interconnection of the various components in terms of a "netlist". A 2-to-1 Multiplexer is a good example to show the interconnection of components. A 2-to-1 MUX consists of one Inverter (G1), two 2-input AND gates (G2 and G3), and one 2-input OR gate (G4). Figure (3) shows how these components are connected to each other. A 2-to-1 MUX has three inputs and one output: two "data inputs", data_0 and data_1, one "sel" and one "output" respectively. The VHDL model of a 2-to-1 MUX is as follows:

```

Line 1. entity 2_1mux is
Line 2. port (data_0, data_1 : in std_logic;
           selector : in std_logic;
           output : out std_logic);
Line 3. end 2_1mux;

--component declaration

Line 4. architecture structure of 2_1mux is
Line 5. component and2_gate
Line 6. port (in_1, in_2: in std_logic;
           output: out std_logic);
Line 7. end component;
Line 8. component or_2gate
Line 9. port (in_1, in_2: in std_logic;
           output: out std_logic);
Line 10. end component;
Line 11. component inv_gate
Line 12. port (in_1: in std_logic;
           output: out std_logic);
Line 13. end component;

--internal signal declaration

Line 14. signal not_sel, x, y: std_logic;

-- not_sel carries signal from G_1 to G_2
-- x carries signal from G_2 to G_4
-- y carries signal from G_3 to G_4

Line 15. begin
      G_1 : inv_gate port map (sel, not_sel);
      G_2 : and2_gate port map (data_0, not_sel, x);
      G_3 : and2_gate port map (data_1, sel, y);
      G_4 : or2_gat port map (x,y,output);
Line 16. end structure
    
```

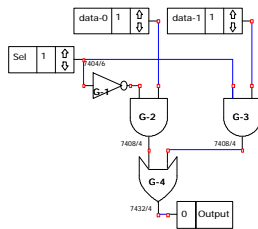


Figure 3. 2-to-1 MUX.

Example 4. Bit-Vector (Array)

A 3-bit binary counter is a good example for a sequential logic circuit with the data type of a bit-vector (an array of three bits). In a 3-bit binary counter, the minterm expression of the three D-flip-flop inputs are as follows:

$$Q(A) = q(A)q(B)' + q(A)q(C)' + q(A)'q(B)q(C)$$

$$Q(B) = q(B) \text{ XOR } q(C)$$

$$Q(C) = q(C)'$$

where “Q” and “q” represent the next state and present state respectively. Figure (4) shows the logic circuit of a 3-bit binary counter and its VHDL model is as follows:

```

Line 1. entity 3bbinary_count is
Line 2. port
      (clk: in bit;
       Z: out bit_vector (2 downto 0));
Line 3. end 3bb_count;
Line 4. architecture structure of 3bb_count is
      type ff_index is (Q(A), Q(B),Q(C) );
      type ff_type is array (ff_index) of bit;
      signal Q: ff-type;
Line 5. begin
      --state: D flip-flops
      Dff: block (clk = '1' and not clk'stable);
      Q(A) <= guarded (q(A) and not q(B)) or (q(A)
and not q(C) or (not q(A) and q(B) and q(C) );
      Q(B) <= guarded q(B) xor q(C);
      Q( C) <= guarded not q(C);
Line 6. end block Dff;
      --output
Line 7. Q <= Q(A) & Q(B) & Q(C);
Line 8. end structure;
    
```

In this example, the statement "ff_index" defines the

indexing scheme for the array type "ff_type". The "block" statement is used to describe level-sensitive synchronous behavior (i.e. clock signals for the gated D-latch control whether the gated latch responds to or ignores transitions on its D inputs. Each "block" statement contains a "guarded" expression which controls the execution of the concurrent signal assignment. The two consecutive dashes "--" in line 6 specifies a "comment".

There are many other good examples that we may find in the variety of VHDL books that are listed in the reference. A good source of information on VHDL is "VHDL International User Forum (VIUF) which is available on the Internet (www.vhdl.org).

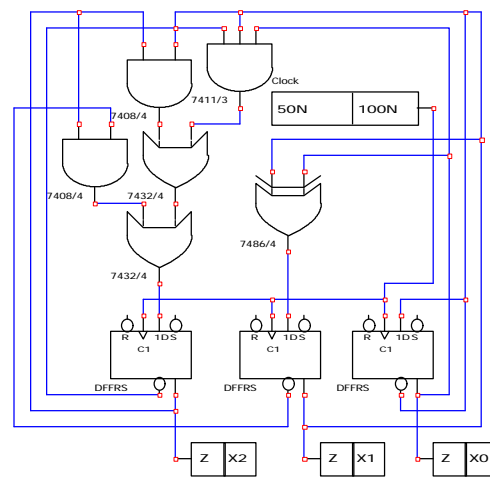


Figure 4. A 3-bit binary counter.

Conclusion:

VHDL is a powerful and popular language for modeling a digital system. Students, with the VHDL knowledge, have a better opportunity to find a well paid position in high-tech companies. If VHDL is not part of a curriculum at the undergraduate level, it is possible to introduce VHDL in a design course such as logic design. Selecting a couple of students design projects and redesigning them by using VHDL will furnish students with the basic knowledge on this subject. There are many channels that student could take to enhance their knowledge after taking such integration of VHDL and logic design course.

References:

- [1] Scarpino, Frank, VHDL and AHDL digital system implementation, Prentice-Hall PTR Pub. Co., 1998.
- [2] Yalamanchili, Sudhakar, VHDL starter's guide, Prentice-Hall Pub. Co., 1998.
- [3] Roth, Jr., Charles H., Digital System Design Using VHDL, PWS Pub. Co., 1998.
- [4] Dewey, Allen. Analysis and Design of Digital Systems with VHDL, PWS Pub. Co., 1997.
- [5] Pellerin David and Taylor, Douglas. VHDL Made Easy!, Prentice-Hall PTR, Pub. Co., 1997.
- [6] Menchini, Paul. Dr. VHDL's Power-Up Guidebook. Check www.VHDLnow.com
- [7] Ashenden, Peter J., The Students Guide to VHDL. Morgan Kaufman Pub. Co.
- [8] Bhasker, J. VHDL Primer, 3rd Ed. Prentice Hall Pub. Co.

Online Resources:

- [1] Cadence, www.cadence.com
- [2] Cypress, www.cypress.com (has a \$99 VHDL synthesis package).
- [3] Mentor Graphic, www.mentor.com
- [4] OrCAD, www.orcad.com (now has joint product with Xilinx).
- [5] Synopsys, www.synopsys.com
- [6] Wiewlogic, www.wielogic.com
- [7] Xilinx, www.xilinx.com
- [8] Synplicity, www.synplicity.com
- [9] comp.land.vhdl (a news group)