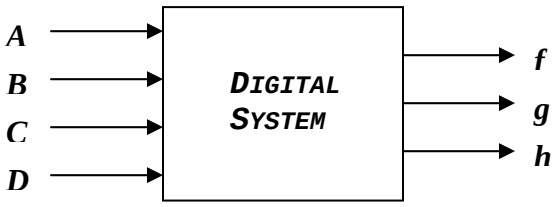
ECE 370: Digital Systems-Logic Design

VHDL Exercise 1: VHDL Concepts and Combinational Circuit Design
Spring 2005

VHDL: A Conceptual Understanding

1. Describe what the purpose of the ENTITY and ARCHITECTURE constructs when designing digital hardware using VHDL. Label the portions of the diagram below that correspond to an ENTITY declaration and an ARCHITECTURE declaration in VHDL.



2. Describe what it means for a VHDL statement to be "concurrent." Give three examples of concurrent statements in VHDL.

3. What is the difference between a structural VHDL design and a behavioral VHDL design? Give two examples of each type of design scheme.

4. What is the general purpose of defining a SIGNAL within an ARCHITECTURE construct? Give two examples of situations that require a SIGNAL definition inside an ARCHITECUTRE body.

5. Given the definition of two multi-bit signals below, what is the internal (compiler) representation of each signal? Use a picture in your explanation and label index numbers in your picture. What is the length of each signal?

```
ARCHITECTURE structure OF myDesign
     SIGNAL x : STD_LOGIC_VECTOR(3 DOWNTO 0);
     SIGNAL y : STD_LOGIC_VECTOR(1 TO 7);
BEGIN
     --more code here...
```

6. Describe the VHDL construct used to design hierarchical digital systems. How does one utilize the particular VHDL construct to implement digital hardware in a hierarchical design scheme?

7. What is the difference between a VHDL COMPONENT and a VHDL PACKAGE? How does one need the other? Why utilize packages if there are already components?

8. What is the purpose of a FOR-GENERATE statement? Give examples of when this construct might be utilized.

9. What is the purpose of the following VHDL packages? Describe how the packages below are related and how they differ from one another. Are these packages used to design in a structural or behavior code using VHDL? Explain your answer.

a) `ieee.std_logic_unsigned.ALL`
b) `ieee.std_logic_signed.ALL`
c) `ieee.std_logic_arith.ALL`

VHDL: Basic Combinational Circuit Design Problems

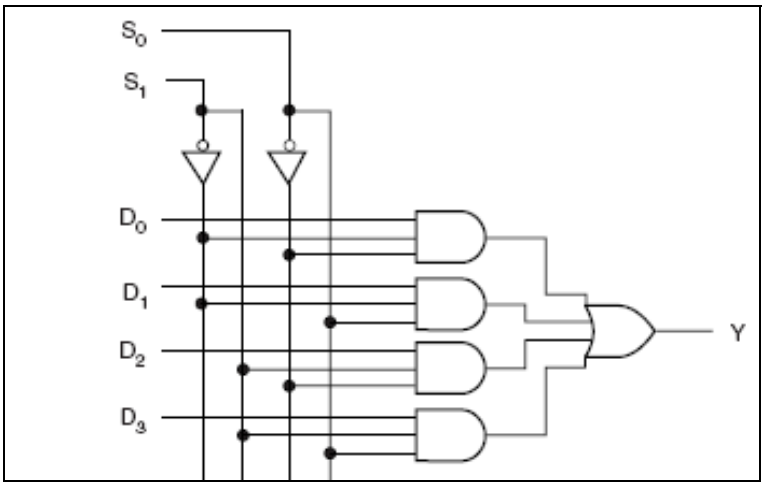
10. Use VHDL to structurally implement the Boolean function *f(A,B,C,D)* given below. Perform both a functional and timing simulation of your design.

$$f(A,B,C,D) = A\overline{B}\,\overline{C} + A\overline{C} + C\overline{D}$$

11. Use VHDL to structurally implement the Boolean function *L(A,B,C)* described by the truth table given below. Perform both a functional and timing simulation of your design.

A	*0 0 0 0*	*1 1 1 1*
B	*0 0 1 1*	*0 0 1 1*
C	*0 1 0 1*	*0 1 0 1*
L(A,B,C)	*1 1 0 1*	*0 0 0 0*

12. Use VHDL to design the circuit below using the following four ways.
    a) Use concurrent signal assignment statements and single-bit signals.
    b) Use concurrent signal assignments and multi-bit signals.
    c) Use a conditional signal assignment and multi-bit signals.
    d) Use a selected signal assignment and multi-bit signals.



13. Design a circuit that inputs a 4-bit 2's complement number *X* and outputs a logic 1 if the number is negative. Design the system using the following four ways.
    a) Use concurrent signal assignment statements and single-bit signals.
    b) Use concurrent signal assignments and multi-bit signals.
    c) Use a conditional signal assignment and multi-bit signals.
    d) Use a selected signal assignment and multi-bit signals.

14. Perform the following sequence of designs.
    a) Design a VHDL entity named `and2` that implements a 2-input AND gate.
    b) Design a VHDL entity named `or2` that implements a 2-input OR gate.
    c) Design a VHDL entity named `inverter` that implements a NOT gate.
    d) Design a VHDL entity named `xor2` that implements a 2-input XOR gate using the above three designs as components. Use as many instances of the components as needed to fully implement the 2-input XOR gate.
    e) Using the above four designs, create a package named `basic_gates_package`. Using the new package, design a 1-bit full adder with as many instances of the individual components as needed.

15. Re-design Problem 11, once using a conditional signal assignment statement and once using a selected signal assignment statement. In each case, define a SIGNAL named `inputs` and concatenate the three input signals together to form a one multi-bit input signal. Based on the new signal, assign to your output accordingly.

16. Use the 1-bit full adder design in Problem 14 as a component in order to implement a 12-bit parallel adder system using a FOR-GENERATE statement and multi-bit input and output signals.

17. Use the 12-bit adder parallel adder design in Problem 16 as a component and design a 12-bit 2's complement adder/subtractor system. Your design should also include overflow-detection logic.

18. Design a 32-bit unsigned adder and 32-bit signed adder system using VHDL behavioral design and any additional VHDL packages.

19. Use behavioral VHDL design to design an 8-bit comparator circuit. This design has two inputs, *X* and *Y*, which are 8-bit 2's complement **signed** numbers. The output specifications are below. Use any additional packages to aid you in your design.

$$L(X,Y) = 1 \text{ if } X < Y, \text{ else } L(X,Y) = 0$$
$$G(X,Y) = 1 \text{ if } X > Y, \text{ else } G(X,Y) = 0$$
$$E(X,Y) = 1 \text{ if } X = Y, \text{ else } E(X,Y) = 0$$