# Homework Assignment

1. [2.6] Write the following statement as structural VHDL code: odd <=(a(0) xor a(1)) xor (a(2) xor a(3)). [Note: you will need to include the unisim library and use its vcomponents package. The source code is available in your Xilinx install directory: C:\Xilinx\14.2\ISE\_DS\ISE\vhdl\src\unisims]
2. [3.1] Write an entity declaration for a memory circuit whose input and output ports are show below. Use only the std\_logic or std\_logic\_vector data types.

* addr: 12-bit address input
* wr\_n: 1-bit write-enable control signal
* oe\_n: 1-bit output-enable control signal
* data: 8-bit bidirectional data bus

1. [3.2] What is the difference between a variable and a signal?
2. [3.3] What is a strongly-typed language?
3. [3.4] What is the limitation of using the bit data type to represent a physical signal?
4. [3.5] Assume that a is a 10-bit signal with the std\_logic\_vector(9 downto 0) data type. List the 10 bits assigned to the a signal.
   1. a <= (others => ‘1’);
   2. a <= (1|3|5|7|9 => ‘1’, others => ‘0’);
   3. a <= (9|7|2 => ‘1’, 6 => ‘0’, 0 => ‘1’, 1|5|8 => ‘0’, 3|4 => ‘0’);
5. [3.6] Assume that a and y are 8-bit signals with the std\_logic\_vector(7 downto 0) data type. If the signals are interpreted as unsigned numbers, the following assignment statement performs a / 8. Explain.  
   y <= “000” & a(7 downto 3);
6. [3.7] Assume the same a as in Problem 3.6. We want to perform a mod 8 and assign the result to y. Rewrite the previous signal assignment statement using only the & operator.
7. [3.11] Determine whether the following signal assignment is syntactically correct. If not, use the proper conversion function and type casting to correct the problem. [see textbook]
8. [3.12] For the following VHDL segment, correct the type mismatch with proper conversion function(s). [see textbook]
9. [3.13] For the following VHDL segment, correct the type mismatch with proper conversion function(s). [see textbook]

# Turn-In Requirements

1. Answers to the above questions
2. Simulation screenshots and source code where appropriate.