Martin Nguyen

4/22/2021

ECE351

**Part 1. (2 pts/each) True/False**

1. SystemVerilog supports Object Oriented programming. **True**
2. "buf" and "notif0" are gate level primitives in SystemVerilog. **True**
3. A primitive gate instance name must be provided when creating an instance of a SystemVerilog primitive gate ((and, or, ...). **True**
4. When creating an instance of a SystemVerilog primitive gate type the output is listed after all of the inputs in the gate description. **False**
5. A value of 'z' on an input or an output in SystemVerilog means that the simulator cannot determine whether the value is a 0 or a 1. **True**
6. Per Roy's definition a SoC contains at least one embedded CPU running an application interfaced to one or more vendor-supplied, 3rd party or custom IP blocks. **False**
7. $monitor() will display the current values of all the variables in its argument list whenever the value of any of the variables changes values. **True**
8. An assignment of 'x or 'z cannot be made to a variable in SystemVerilog. **True**
9. You may use the .\* shorthand notation only when all of the variable names match all of the port names in an instantiation of a module. **False**
10. The following Verilog code is valid:

wire [7:0] a, b;

wire [0:15] sum = a + b

**True**

1. The default type for an enum is int. **True.**
2. The following Verilog code snippet will result in an 8-bit tri-state buffer:

wire [7:0] data\_out, data\_in;

wire en;

assign data\_out = en ? data\_in : 1’bz;

**True**

1. Assigning a 4-state variable to a 2-state variable in a simulation results causes an error message to be displayed on the console. **True**
2. x = ^8’b11100011 will be equal to 1 if it is simulated in System Verilog. **True**
3. localparams in a module can be overridden when the module is instantiated. **False**

**Part 2. (3pts/each) Short answer**

1. What are a few similarities and differences between a SystemVerilog int and a SystemVerilog integer data type?

* Differences: The difference between an integer and in is that an integer is a 4-state type while an int is a 2-state type.
* Similarities: The similarity is that they both use for 32 bit signed integer.

1. Why do we need to be cautious when we assign a 4-state variable to a 2-state variable?

* When we convert a 4-state to 2-state any unknown or high-impedance bits shall be converted to zero.

1. Answer these questions about Synthesis? Keep your answers consice (2 or 3 sentences for each question):
   1. What is Synthesis?

* What is synthesis is the process which Verilog HDL models is being convert to a gate-level implementations and maps these to the target hardware/technology.
  1. What information would you be likely to include in a constraints file?
* What information would you be having in your constraint file is the pin that is use on a given devices/board, allocation of a clock resource and speed it run at, the internal feature of certain devices.
  1. Even though you can write RTL code to be largely technology dependent, Synthesis ties your code more closely to a specific target technology. Why is this?
* Because when synthesis your RTL code get translate to a gate and optimize the logic, which make the RTL code that are being described in Verilog get tied to a particular clock cycle of specific target. The synthesis netlist has the same clock to clock behavior that will help the RTL code to be used again and again at gate simulation level.

1. Define the follow terms. Keep your definition for each term to a few sentences:
   1. Module: A block of Verilog code that implements a certain function. A module can be implemented within other modules and a high level module can communicate with input/output ports.
   2. Instance of a module: It when you create a variable from the module and call the module with data being pass in as the parameter.
   3. Port: It a set of signals that act as inputs or output to a particular module and are the primary way of communication between the modules.
2. Explain the difference between the data types of logic, reg, and wire.

* Logic: Usually infers a general purpose var logic 4-state variable of a user-defined vector size, except on module input/output ports, where wire logic inferred.
* Reg: an obsolete general purpose 4-state variable of a user-defined vector size; equivalent to var logic.
* Wire: these are a type that connect the ports to one another which make them like busses in way where they pass data from one end of the port to another.

**Part 3. Multiple choice**

1. (Circle the best answer, no partial credit) The following System Verilog code snippet

logic [3:0] a = 4'hFF;  
initial begin  
     $display("%d", a);  
     a = '1;  
    $display("%d", a);  
end

displays the following:

a. 15  
    1  
b. 15  
    15  
c. 255  
   1  
d. 255  
    255  
e. None of the above

**YOUR ANSWER: C**

2. (Circle the best answer: no partial credit) In the statement assign r = |6’b0x1x01,

what is the value of r after execution?

a. r = 6’b0x1x01

b. r = 1’b1

c. r = 6’b0

d. r = 1’bx

e. r = 6’bxxxxxx

**YOUR ANSWER: B**

3. (Circle the best answer) The following SystemVerilog code snippet

logic [7:0] b1 = 8'b1101zzzz;

byte b2;

b2 = b1;

$write(“b1=%b, ”,b1);

$display(“b2=%h”,b2);

will display:

a. b1 11-1xxxx, b2 = d0

b. b1 = 8'b1101zzzz, b2 = dz

c. b1 = 1101zzzz, b2 = xx

d. b1 = 1101zzzz, b2=d0

e. None of the above

**YOUR ANSWER: D**

4. (Circle all of the items that are true, score is right - wrong) What is the difference between

bit[7:0] sig\_1;

byte sig\_2;

a. both sig\_1 and sig\_2 are 8-bit 2-state variables

b. both sig\_1 and sig\_2 are treated as signed numbers

c. sig\_1 is treated as a signed number but sig\_2 is treated as unsigned

d. sig\_1 and sig\_2 have a different number of bits

e. bit[7:0] is synthesizable but byte is not synthesizable

**YOUR ANSWER: C, D, E**

5. (Circle all of the items that are true, score is right - wrong) The following can be said about port

declarations in a module:

a. Both input and output ports can be of type wire or logic

b. The following port list is valid:

module alu (a, b, result);

c. The default size of reg, logic, bit, and wire is 1 bit wide

d. The port type and data type can be a variable or any of the net types and data types

e. The default type when no data type is specified is logic

**YOUR ANSWER: A, E, C**