SC1005 Digital Logic
Tutorial 7

Q1. A traffic light for a toy car track has a 3-bit one-hot vector input, with MSB corresponding to red and the LSB to green. However, the controller delivered with the track has only a 2-bit output for the lights, encoded as: red = “00”, yellow = “01” and green = “10”. The controller outputs “11” when there are no lights active.

Design a Verilog interface (using assign statements) to go between the controller and the lights.

Q2. A digital thermostat has two 8-bit unsigned binary inputs representing the target temperature and the actual temperature in degrees centigrade (°C). The thermostat has two outputs: one to turn a heater on when the actual temperature is 4°C below the target, and one to turn a cooler on when the actual temperature is 4°C above the target

(a) Design a Verilog module, *thermo*, with two 8-bit inputs, Tset and Tact and two 1-bit outputs Hon and Con. Use a parameter statement to specify the 8-bit width.

(b) The module designed in part (a) is instantiated in another module. Write the Verilog statement to instantiate the thermo module with identifier U1 using the same signal names in the upper module.

(c) Part way through the design process, the design team finds out that the temperature sensor and the target set-point both have 12-bit outputs. Describe a simple change to the Verilog statement to instantiate the thermo module so that this will not affect the operation.

Q3. The following Verilog module has a number of mistakes/errors that would result in it synthesizing incorrectly.

(a) Identify the mistakes/errors and rewrite the module to correct them.

(b) How could you rewrite that module using a single conditional assignment?

```
module thingamajig (input [3:0] a, b, c,
                    input [1:0] sel,
                    output [3:0] result);
      always @ (a,b)
      begin
         case (sel)
            2’b00 : result = a;
            2’b01 : result = b;
            2’b10 : result = c;
         endcase
      end
endmodule
```

result = 0 ;

reg 3

case

```
assign result = (sel==2'b00)? a:
                (sel==2'b01)? b:c;
```

b)

```
assign result = sel [1]? (sel [0] ? 0: c) :
                (sel [0] ? b : a);
```

Q4. (a) Write a Verilog module that implements a 3-to-8 decoder using a case statement in an always block.

(b) What modification would you make in order to add an enable (en) to the circuit?