# 3 Verilog

Please answer the following four questions about Verilog.

(a) [10 points] Does the following code result in a sequential circuit or a combinational circuit? Please explain why.

```
module one (input clk, input a, input b, output reg [1:0] q);
 always @ (*)
   if (b)
      q <= 2'b01;
   else if (a)
      q <= 2'b10;
endmodule
```

Answer and concise explanation:

This code results in a sequential circuit because a latch is required to store the old value of `q` if both conditions are **not** satisfied.

(b) [10 points] What is the value of the output `z` if the input `c` is `10101111` and `d` is `01010101`?

```
module two (input [7:0] c, input [7:0] d, output reg [7:0] z);
always @ (c,d)
  begin
    z = 8'b00000001;
    z[7:5] = c[5:3];
    z[4] = d[7];
    z[3] = d[7];
  end
endmodule
```

Please answer below. Show your work.

10100001. Last assignment of a bit overrides all previous assignments.

(c) [10 points] Is the following code correct? If not, please explain the mistake and how to fix it.

```
1  module mux2 ( input [1:0] i, input sel, output z);
2    assign z= (sel) ? i[1]:i[0];
3  endmodule
4
5  module three ( input [3:0] data, input sel1, input sel2, output z);
6
7    wire m;
8
9    mux2 i0 (.i(data[1:0]), .sel(sel1), .z(m[0]) );
10   mux2 i1 (.i(data[3:2]), .sel(sel1), .z(m[1]) );
11   mux2 i2 (.i(m), .sel(sel2), .z(z) );
12
13 endmodule
```

Answer and concise explanation:

No. The wire m is declared to be only 1-bit wide but it needs to be 2-bit wide.

(d) [10 points] Does the following code correctly implement a multiplexer?

```
1  module four (input sel, input [1:0] data, output reg z);
2    always@(sel)
3    begin
4      if(sel == 1'b0)
5        z = data[0];
6      else
7        z = data[1];
8    end
9  endmodule
```

Answer and concise explanation:

No. The input `data` is not in the sensitivity list and therefore changes to the input would not be reflected in the output `z`.