4. In this question for each part there will be two Verilog code snippets. For each part you will have to say whether both, only one, or none of the code snippets fulfill what is being asked. All code snippets are syntactically correct. They will compile and produce either a sequential circuit or a combinational circuit.

   (a) (2 points) Which code snippet generates a sequential circuit?

(A)

```
module a1 (input b,
           output reg c);
  reg a;
  always @ (*)
  begin
    a = b;
    c = ~a;
  end
endmodule
```

(B)

```
module a2 (input clk,
           input rst,
           input d,
           output reg q);
always @ (clk, rst, d)
   if (rst)
      q <= 1'b0;
   else if (clk)
      q <= d ;
   else
      q <= ~d;
endmodule
```

√ **none** ◯ A ◯ B ◯ Both A and B

   (b) (2 points) Which code snippet properly instantiates the module mux2?

```
module mux2 (input d1, input d2, input s, output out);
 assign out = s? d1:d2;
endmodule
```

(A)

```
module b1 (input a,
           input b,
           input sel,
           output q);
  mux2 first (a,b,sel,q);
endmodule
```

(B)

```
module b2 (input a,
           input b,
           input sel,
           output q);
  mux2 second (.d1(a),
               .d2(b),
               .s(sel),
               .out(q)
              );
endmodule
```

◯ none ◯ A ◯ B √ **Both A and B**

(c) (2 points) Which code snippet results in a 2-input multiplexer ?

(A)

```
module c1 (input sel,
           input a,
           input b,
           output z);
 assign z = (sel) ? a : 0;
endmodule
```

(B)

```
module c2 (input sel,
           input a,
           input b,
           output z);
 assign z = ~sel & b | sel & a;
endmodule
```

◯ none ◯ A √ **B** ◯ Both A and B

(d) (2 points) Which code snippet(s) will produce a 8-bit value which is composed of (from MSB to LSB), $c_3c_2c_1d_6d_6110$ (c and d are both 8-bit values)?

(A)

```
module d1 (input [7:0] c,
           input [7:0] d,
           output [7:0] z);
  assign z = {c[3:1],
              {2{d[6]}},
              3'b110    };
endmodule
```

(B)

```
module d2 (input [7:0] c,
           input [7:0] d,
           output reg [7:0] z);
always @ (c,d)
  begin
    z<= 8'b00000110;
    z[7:5] <= c[3:1];
    z[4] <= d[6];
    z[3] <= d[6];
  end
endmodule
```

◯ none ◯ A ◯ B √ **Both A and B**

(e) (2 points) Which code snippets produce a combinational circuit?

(A)

```
module e1 (input clk,
           input a,
           input b,
           output reg [1:0] q);
  always @ (clk)
    if (clk)
      q <= 2'b10;
    else
      q <= 2'b01;
endmodule
```

(B)

```
module e2 (input clk,
           input a,
           input b,
           output reg [1:0] q);
 always @ (*)
   if (a)
      q <= 2'b01;
   else if (b)
      q <= 2'b10;

endmodule
```

◯ none √ **A** ◯ B ◯ Both A and B
