

```

1  /*
2  Ben Davis
3  2/2/24
4  EE 371
5  Lab 3, Task 2
6
7  This is a module that takes in eight inputs. Four of them
8  are for the start and end coordinates of the beginning of
9  the animation, and the other four are for the end coordinates
10 of the animation. This module is for task 2, so it has only
11 been programmed to draw a straight line moving.
12
13 */
14
15 module animator (
16     input logic clk, reset,
17     input logic [9:0] x0, x1, w0, w1, //all x coordinates
18     output logic [9:0] a0, a1 //next x coordinates
19 );
20
21     int xw0_step;
22     int xw1_step;
23
24     assign xw0_step = (w0 > x0) ? 1 : -1;
25     assign xw1_step = (w1 > x1) ? 1 : -1;
26
27     always_ff @(posedge clk) begin
28         if (reset) begin
29             a0 <= x0;
30             a1 <= x1;
31         end else begin
32
33             //expanding to the left
34             if(a0 > w0) begin
35                 a0 <= a0 - 1;
36             end else begin
37                 a0 <= a0;
38             end
39
40             //expanding to the right
41             if(a1 < w1) begin
42                 a1 <= a1 + 1;
43             end else begin
44                 a1 <= a1;
45             end
46         end
47     end //of ff
48
49 endmodule
50 //testbench
51 module animator_testbench();
52
53     logic clk, reset;
54     logic [9:0] x0, x1, w0, w1, a0, a1;
55
56     animator dut (.clk, .reset, .x0, .x1, .w0, .w1,
57                 .a0, .a1);
58
59     parameter clk_p = 100;
60     initial begin
61         clk <= 0;
62         forever #(clk_p / 2) clk <= ~clk;
63     end
64
65     //testing the instance of drawing a horizontal line that
66     //is always at y = 80, and moves from an initial x 200 to
67     //200 to a line that spans 100 to 300
68     initial begin
69         reset <= 1; x0 <= 200; x1 <= 200; w0 <= 100; w1 <= 300;
70         @(posedge clk);
71         reset <= 0; @(posedge clk);
72         @(posedge clk);
73         @(posedge clk);

```

```
74         @(posedge clk);
75         @(posedge clk);
76         @(posedge clk);
77         $stop;
78     end
79 endmodule //testbench
```