```
1   // A driver for the 16x16x2 LED display expansion board.
2   // Read below for an overview of the ports.
3   // IMPORTANT: You do not need to necessarily modify this file. But if you do, be sure you
    know what you are doing.
4
5   // FREQDIV: (Parameter) Sets the scanning speed (how often the display cycles through rows)
6   //          The CLK input divided by 2^(FREQDIV) is the interval at which the driver
    switches rows.
7   // GPIO_1: (Output) The 36-pin GPIO1 header, as on the DE1-SoC board.
8   // RedPixels: (Input) A 16x16 array of logic items corresponding to the red pixels you'd
    like to have lit on the display.
9   // GrnPixels: (Input) A 16x16 array of logic items corresponding to the green pixels you'd
    like to have lit on the display.
10  // EnableCount: (Input) Whether to continue moving through the rows.
11  // CLK: (Input) The system clock.
12  // RST: (Input) Resets the display driver. Required during startup before use.
13  module LEDDriver #(parameter FREQDIV = 0) (GPIO_1, RedPixels, GrnPixels, EnableCount, CLK,
    RST);
14      output logic [35:0] GPIO_1;
15      input logic [15:0][15:0] RedPixels ;
16      input logic [15:0][15:0] GrnPixels ;
17      input logic EnableCount, CLK, RST;
18
19      reg [(FREQDIV + 3):0] Counter;
20      logic [3:0] RowSelect;
21      assign RowSelect = Counter[(FREQDIV + 3):FREQDIV];
22
23      always_ff @(posedge CLK)
24      begin
25          if(RST) Counter <= 'b0;
26          if(EnableCount) Counter <= Counter + 1'b1;
27      end
28
29      assign GPIO_1[35:32] = RowSelect;
30      assign GPIO_1[31:16] = { GrnPixels[RowSelect][0], GrnPixels[RowSelect][1], GrnPixels[
    RowSelect][2], GrnPixels[RowSelect][3], GrnPixels[RowSelect][4], GrnPixels[RowSelect][5],
    GrnPixels[RowSelect][6], GrnPixels[RowSelect][7], GrnPixels[RowSelect][8], GrnPixels[
    RowSelect][9], GrnPixels[RowSelect][10], GrnPixels[RowSelect][11], GrnPixels[RowSelect][12],
     GrnPixels[RowSelect][13], GrnPixels[RowSelect][14], GrnPixels[RowSelect][15] };
31      assign GPIO_1[15:0] = { RedPixels[RowSelect][0], RedPixels[RowSelect][1], RedPixels[
    RowSelect][2], RedPixels[RowSelect][3], RedPixels[RowSelect][4], RedPixels[RowSelect][5],
    RedPixels[RowSelect][6], RedPixels[RowSelect][7], RedPixels[RowSelect][8], RedPixels[
    RowSelect][9], RedPixels[RowSelect][10], RedPixels[RowSelect][11], RedPixels[RowSelect][12],
     RedPixels[RowSelect][13], RedPixels[RowSelect][14], RedPixels[RowSelect][15] };
32  endmodule
33
34  module LEDDriver_Test();
35      logic CLK, RST, EnableCount;
36      logic [15:0][15:0]RedPixels;
37      logic [15:0][15:0]GrnPixels;
38      logic [35:0] GPIO_1;
39
40      LEDDriver #(.FREQDIV(2)) Driver(.GPIO_1, .RedPixels, .GrnPixels, .EnableCount, .CLK, .
    RST);
41
42      initial
43      begin
44          CLK <= 1'b0;
45          forever #50 CLK <= ~CLK;
46      end
47
48      initial
49      begin
50          EnableCount <= 1'b0;
51          RedPixels <= '{default:0};
52          GrnPixels <= '{default:0};
53          @(posedge CLK);
54
55          RST <= 1; @(posedge CLK);
56          RST <= 0; @(posedge CLK);
57          @(posedge CLK); @(posedge CLK); @(posedge CLK);
58
59          GrnPixels[1][1] <= 1'b1; @(posedge CLK);
```

```
60            EnableCount <= 1'b1; @(posedge CLK); #1000;
61            RedPixels[2][2] <= 1'b1;
62            RedPixels[2][3] <= 1'b1;
63            GrnPixels[2][3] <= 1'b1; @(posedge CLK); #1000;
64            EnableCount <= 1'b0; @(posedge CLK); #1000;
65            GrnPixels[1][1] <= 1'b0; @(posedge CLK);
66            $stop;
67
68        end
69    endmodule
70
71    module LEDDriver_TestPhysical(CLOCK_50, RST, Speed, GPIO_1);
72        input logic CLOCK_50, RST;
73        input logic [9:0] Speed;
74        output logic [35:0] GPIO_1;
75        logic [15:0][15:0]RedPixels;
76        logic [15:0][15:0]GrnPixels;
77        logic [31:0] Counter;
78        logic EnableCount;
79
80        LEDDriver #(.FREQDIV(15)) Driver (.CLK(CLOCK_50), .RST, .EnableCount, .RedPixels, .
      GrnPixels, .GPIO_1);
81
82        //                          F E D C B A 9 8 7 6 5 4 3 2 1 0
83        assign RedPixels[00] = '{1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1};
84        assign RedPixels[01] = '{1,1,0,0,0,0,0,0,0,0,0,0,0,0,1,1};
85        assign RedPixels[02] = '{1,0,1,1,1,1,1,1,1,1,1,1,1,1,0,1};
86        assign RedPixels[03] = '{1,0,1,1,0,0,0,0,0,0,0,0,1,1,0,1};
87        assign RedPixels[04] = '{1,0,1,0,1,1,1,1,1,1,1,1,0,1,0,1};
88        assign RedPixels[05] = '{1,0,1,0,1,1,0,0,0,0,1,1,0,1,0,1};
89        assign RedPixels[06] = '{1,0,1,0,1,0,1,1,1,1,0,1,0,1,0,1};
90        assign RedPixels[07] = '{1,0,1,0,1,0,1,0,1,1,0,1,0,1,0,1};
91        assign RedPixels[08] = '{1,0,1,0,1,0,1,1,0,1,0,1,0,1,0,1};
92        assign RedPixels[09] = '{1,0,1,0,1,0,1,1,1,1,0,1,0,1,0,1};
93        assign RedPixels[10] = '{1,0,1,0,1,1,0,0,0,0,1,1,0,1,0,1};
94        assign RedPixels[11] = '{1,0,1,0,1,1,1,1,1,1,1,1,0,1,0,1};
95        assign RedPixels[12] = '{1,0,1,1,0,0,0,0,0,0,0,0,1,1,0,1};
96        assign RedPixels[13] = '{1,0,1,1,1,1,1,1,1,1,1,1,1,1,0,1};
97        assign RedPixels[14] = '{1,1,0,0,0,0,0,0,0,0,0,0,0,0,1,1};
98        assign RedPixels[15] = '{1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1};
99
100       assign GrnPixels[00] = '{1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1};
101       assign GrnPixels[01] = '{0,1,1,1,1,1,1,1,1,1,1,1,1,1,1,0};
102       assign GrnPixels[02] = '{0,1,1,0,0,0,0,0,0,0,0,0,0,1,1,0};
103       assign GrnPixels[03] = '{0,1,0,1,1,1,1,1,1,1,1,1,1,0,1,0};
104       assign GrnPixels[04] = '{0,1,0,1,1,0,0,0,0,0,0,1,1,0,1,0};
105       assign GrnPixels[05] = '{0,1,0,1,0,1,1,1,1,1,1,0,1,0,1,0};
106       assign GrnPixels[06] = '{0,1,0,1,0,1,1,0,0,1,1,0,1,0,1,0};
107       assign GrnPixels[07] = '{0,1,0,1,0,1,0,1,0,0,1,0,1,0,1,0};
108       assign GrnPixels[08] = '{0,1,0,1,0,1,0,0,1,0,1,0,1,0,1,0};
109       assign GrnPixels[09] = '{0,1,0,1,0,1,1,0,0,1,1,0,1,0,1,0};
110       assign GrnPixels[10] = '{0,1,0,1,0,1,1,1,1,1,1,0,1,0,1,0};
111       assign GrnPixels[11] = '{0,1,0,1,1,0,0,0,0,0,0,1,1,0,1,0};
112       assign GrnPixels[12] = '{0,1,0,1,1,1,1,1,1,1,1,1,1,0,1,0};
113       assign GrnPixels[13] = '{0,1,1,0,0,0,0,0,0,0,0,0,0,1,1,0};
114       assign GrnPixels[14] = '{0,1,1,1,1,1,1,1,1,1,1,1,1,1,1,0};
115       assign GrnPixels[15] = '{1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1};
116
117       always_ff @(posedge CLOCK_50)
118       begin
119           if(RST) Counter <= 'b0;
120           else
121           begin
122               Counter <= Counter + 1'b1;
123               if(Counter >= Speed)
124               begin
125                   EnableCount <= 1'b1;
126                   Counter <= 'b0;
127               end
128               else EnableCount <= 1'b0;
129           end
130       end
131   endmodule
```