```
1   /*-------------------------------------------------------------------------------
2   Name Lamin Jammeh
3   CLass: EE417 Summer 2024
4   Lesson 07 HW Question 2
5   Group: Ron Kalin/ Lamin Jammeh
6   Project Description: Datapath module takes comman from controoler module and counts
7   use a one hot count by shifting the value 1 left and right in an 8 bit input
8   -------------------------------------------------------------------------------*/
9
10  /*------------up/down count-------------------------------*/
11  module DataPath #(parameter WIDTH = 8, parameter CYCLES = 18) (
12      input clk,
13      input reset,
14      input shift_left,
15      input shift_right,
16      input load,
17      input [WIDTH-1:0] data_in,
18      output reg [WIDTH-1:0] count,
19      output reg done
20  );
21
22  reg [4:0] shift_reg; // 5-bit counter to count up to 18 CYCLES
23
24  always @ (posedge clk or posedge reset)
25  begin
26      //condition for reset
27      if (reset)
28      begin
29          count <= 8'b00000001; // output count at reset is the first state of the counter
30          shift_reg <= 0;
31          done <= 0;
32      end
33      //condition to start loading the output
34      else if (load)
35      begin
36          count <= data_in; // count is still 1 when load comes on
37          shift_reg <= 1;
38          done <= 0;
39      end
40      //condition to shift 1 left and right in count
41      else if (shift_left || shift_right)
42      begin
43          if (shift_reg == CYCLES)
44          begin
45              shift_reg <= 1;
46              count <= 8'b00000001;
47              //done <= 1;
48          end
49          else begin
50              shift_reg <= shift_reg + 1;            //all start conditions are active,
    increment shift_reg
51              //define a condition for counting up
52              if (shift_left)
53              begin
54                 count <= {count[WIDTH-2:0], count[WIDTH-1]}; // Shift left by joining
    count[6:0] to count[7]
55              end
56              else if (shift_right)
57              begin
58                 count <= {count[0], count[WIDTH-1:1]}; // Shift right by joining count[0] to
    count[7:1]
59              end
60          end
61          done <= (shift_reg == CYCLES);
62      end
63  end
64
65  endmodule
66
```