```
 1 //-------------------------------------------------------------
 2 //
 3 //  Top Level Verilog function
 4 //                        By:  Kenneth Maxon 08/08/04
 5 //
 6 //-------------------------------------------------------------
 7 module enc_cpld(
 8             input wire sys_clock,
 9             output wire spi_miso,
10             input wire spi_sck,
11             input wire spi_cs,
12             input wire encoder_1a,
13             input wire encoder_1b,
14             input wire encoder_2a,
15             input wire encoder_2b
16             );
17
18 //-------------------- local variables
19 parameter count_width = 8;  // this one configurable parameter sets how
20                             // many bits will be counted and 2x this value
21                             // will be transmitted.  Ex: setting 8 gives
22                             // both encoders a maximum count of 255 and
23                             // transmits 16 bits for the spi request.
24 wire [count_width - 1:0] enc1_data_out;
25 wire [count_width - 1:0] enc2_data_out;
26
27 //-------------------- Instantiate the functions
28 spi_slave #(count_width << 1) my_spi (
29             .sys_clock(sys_clock),
30             .spi_miso(spi_miso),
31             .spi_sck(spi_sck),
32             .spi_cs(spi_cs),
33             .outgoing_data({enc1_data_out[count_width - 1:0],enc2_data_out[count_width - 1:0]})
34             );
35
36 encoder_func #(count_width) enc1(
37             .sys_clock(sys_clock),
38             .encoder_data_out(enc1_data_out[count_width - 1:0]),
39             .encoder_a(encoder_1a),
40             .encoder_b(encoder_1b)
41             );
42
43 encoder_func #(count_width) enc2(
44             .sys_clock(sys_clock),
45             .encoder_data_out(enc2_data_out[count_width - 1:0]),
46             .encoder_a(encoder_2a),
47             .encoder_b(encoder_2b)
48             );
49
50 endmodule
51
52
```