```
1   //
2   //  LEDG :  To turn off, write "0"
3   //          To turn on,  write "1"
4   //
5   //  HEX :   To turn off, write "1"
6   //          To turn on, write "0"
7   //
8   //            _0_
9   //         5|_6_|1
10  //         4|___|2
11  //            3
12  //
13  //  BUTTON : Push --> "0"
14  //           Release --> "1"
15  //
16  //  SW: Down (towards the edge of the board)  --> "0"
17  //      Up --> "1"
18  //
19  `timescale 1ns/1ps
20
21  module MIPS_System(
22    input         CLOCK_50,
23    input   [2:0] BUTTON,
24    input   [9:0] SW,
25    output  [6:0] HEX3_D,
26    output  [6:0] HEX2_D,
27    output  [6:0] HEX1_D,
28    output  [6:0] HEX0_D,
29    output  [9:0] LEDG);
30
31    wire reset;
32    wire reset_poweron;
33    reg  reset_ff;
34    wire clk0;
35    wire locked;
36    wire [31:0] inst_addr;
37    wire [31:0] inst;
38    wire [31:0] data_addr;
39    wire [31:0] write_data;
40    wire [31:0] read_data_timer;
41    wire [31:0] read_data_uart;
42    wire [31:0] read_data_gpio;
43    wire [31:0] read_data_mem;
44    reg  [31:0] read_data;
45    wire        cs_mem_n;
46    wire        cs_timer_n;
47    wire        cs_gpio_n;
48    wire        data_we;
49
50    wire clk90;
51    wire clk180;
52    wire data_re;
53
54    // reset =  BUTTON[0]
55    // if BUTTON[0] is pressed, the reset goes down to "0"
56    // reset is a low-active signal
57    assign  reset_poweron = BUTTON[0];
58    assign  reset = reset_poweron & locked;
59
60    always @(posedge clk0)  reset_ff <= reset;
61
62    ALTPLL_clkgen pll0(
63            .inclk0   (CLOCK_50),
64            .c0       (clk0),
65            .c1       (clk90),
66            .c2       (clk180),
```

```
67              .locked   (locked));
68
69
70      always @(*)
71      begin
72         if      (~cs_timer_n) read_data <= read_data_timer;
73         else if (~cs_gpio_n)  read_data <= read_data_gpio;
74         else                  read_data <= read_data_mem;
75      end
76
77
78      mips mips_cpu (
79          .clk           (clk0),
80          .reset         (~reset_ff),
81          .pc            (inst_addr),
82          .instr         (inst),
83          .memwrite      (data_we),  // data_we: active high
84          .memaddr       (data_addr),
85          .memwritedata  (write_data),
86          .memreaddata   (read_data));
87
88       assign data_re = ~data_we;
89
90       // Port A: Instruction
91       // Port B: Data
92       ram2port_inst_data Inst_Data_Mem (
93          .address_a   (inst_addr[12:2]),
94          .address_b   (data_addr[12:2]),
95          .byteena_b   (4'b1111),
96          .clock_a     (clk90),
97          .clock_b     (clk180),
98          .data_a      (),
99          .data_b      (write_data),
100         .enable_a    (1'b1),
101         .enable_b    (~cs_mem_n),
102         .wren_a      (1'b0),
103         .wren_b      (data_we),
104         .q_a         (inst),
105         .q_b         (read_data_mem));
106
107
108     Addr_Decoder Decoder (
109         .Addr        (data_addr),
110         .CS_MEM_N    (cs_mem_n) ,
111         .CS_TC_N     (cs_timer_n),
112         .CS_UART_N   (),
113         .CS_GPIO_N   (cs_gpio_n));
114
115
116     TimerCounter Timer (
117         .clk     (clk0),
118         .reset   (~reset_ff),
119         .CS_N    (cs_timer_n),
120         .RD_N    (~data_re),
121         .WR_N    (~data_we),
122         .Addr    (data_addr[11:0]),
123         .DataIn  (write_data),
124         .DataOut (read_data_timer),
125         .Intr    ());
126
127
128      GPIO uGPIO (
129       .clk    (clk0),
130       .reset  (~reset_ff),
131       .CS_N   (cs_gpio_n),
132       .RD_N   (~data_re),
```

```
133       .WR_N   (~data_we),
134       .Addr   (data_addr[11:0]),
135       .DataIn (write_data),
136       .DataOut(read_data_gpio),
137       .Intr   (),
138       .BUTTON (BUTTON[2:1]),
139       .SW     (SW),
140       .HEX3   (HEX3_D),
141       .HEX2   (HEX2_D),
142       .HEX1   (HEX1_D),
143       .HEX0   (HEX0_D),
144       .LEDG   (LEDG));
145
146   endmodule
147
```