```
 1  `timescale 1ns / 1ps
 2  /**********************************************************************************
 3   *
 4   * Author:   Jesus Luciano
 5   * Filename: MIPS_32.v
 6   * Date:     1/22/2019
 7   * Version:  1.0
 8   *
 9   * Notes:    This module does all operations except multiply and divide
10   *           Y_lo is a function of the 2 32-bit inputs and FS
11   *           Y_hi is set to all zeros for this module
12   *           Status flags are set appropriately based on FS, inputs and outputs
13   *           Status flags are set to 'x' if not affected by operation
14   *
15   **********************************************************************************/
16  module MIPS_32(S, T, FS, V, C, Y_hi, Y_lo);
17
18      //delcare inputs
19      input  [31:0]  S, T;
20      input  [ 4:0] FS;
21
22      //delcare outputs
23      //only overflow and carry flag are computer in this module
24      output reg           V,    C;
25      output reg [31:0] Y_hi, Y_lo;
26
27      //integer declaration for slt
28      integer intS, intT;
29
30      //declare parameters for instructions
31      parameter PASS_S = 5'h00, PASS_T = 5'h01, ADD = 5'h02,
32                ADDU= 5'h03, SUB  = 5'h04, SUBU = 5'h05,
33                SLT = 5'h06, SLTU = 5'h07, AND  = 5'h08,
34                OR  = 5'h09, XOR  = 5'h0A, NOR  = 5'h0B,
35                SRL = 5'h0C, SRA  = 5'h0D, SLL  = 5'h0E,
36                ANDI= 5'h16, ORI  = 5'h17, LUI  = 5'h18,
37                XORI= 5'h19, INC  = 5'h0F, INC4 = 5'h10,
38                DEC = 5'h11, DEC4 = 5'h12, ZEROS= 5'h13,
39                ONES= 5'h14, SP_INIT = 5'h15;
40
41      always @ (*)  // S T FS - verilog will interperet as unsigned
42         case(FS)
43             //arithmetic
44             PASS_S : {V, C, Y_hi, Y_lo} = {2'bx, 32'h0, S};
45             PASS_T : {V, C, Y_hi, Y_lo} = {2'bx, 32'h0, T};
46
47             ADD    : begin//signed
48                         Y_hi = 32'b0; {C, Y_lo} = S + T;
49                         V = (~S[31] & ~T[31] &  Y_lo[31]) |//pos+pos=neg
50                             ( S[31] &  T[31] & ~Y_lo[31]); //neg+neg=pos
51                      end
52             ADDU   : begin //unsigned
53                         Y_hi = 32'b0; {C, Y_lo} = S + T;
54                         V = C;
55                      end
56             SUB    : begin//signed
57                         Y_hi = 32'b0; {C, Y_lo} = S - T;
```

```
58                          V = (~S[31] &  T[31] &  Y_lo[31]) |//pos-neg=neg
59                              ( S[31] & ~T[31] & ~Y_lo[31]); //neg-pos=pos
60                        end
61              SUBU    : begin //sub unsigned
62                          Y_hi = 32'b0; {C, Y_lo} = S - T;
63                          V = C;//overflow is the same as carry
64                        end
65              SLT     : begin//both inputs cast to integers
66                          Y_hi = 32'b0; {V,    C} = 2'bx;
67                          intS = S; intT = T;
68                          Y_lo = (intS < intT) ? 32'h1 : 32'h0;
69                        end
70              SLTU    : {V, C, Y_hi, Y_lo} = {2'bx, 32'h0, (S < T) ? 32'h1 : 32'h0};
71
72              //logical
73              AND     : {V, C, Y_hi, Y_lo} = {2'bx, 32'h0,  S & T };
74              OR      : {V, C, Y_hi, Y_lo} = {2'bx, 32'h0,  S | T };
75              XOR     : {V, C, Y_hi, Y_lo} = {2'bx, 32'h0,  S ^ T };
76              NOR     : {V, C, Y_hi, Y_lo} = {2'bx, 32'h0,~(S | T)};
77              SRL     : {V, C, Y_hi, Y_lo} = {1'bx, T[0] , 32'h0,  1'b0,  T[31:1]};
78              SRA     : {V, C, Y_hi, Y_lo} = {1'bx, T[0] , 32'h0,  T[31], T[31:1]};
79              SLL     : {V, C, Y_hi, Y_lo} = {1'bx, T[31], 32'h0,  T[30:0], 1'b0 };
80              ANDI    : {V, C, Y_hi, Y_lo} = {2'bx, 32'h0, S & {16'h0, T[15:0]}  };
81              ORI     : {V, C, Y_hi, Y_lo} = {2'bx, 32'h0, S | {16'h0, T[15:0]}  };
82              LUI     : {V, C, Y_hi, Y_lo} = {2'bx, 32'h0, T[15:0], 16'h0};
83              XORI    : {V, C, Y_hi, Y_lo} = {2'bx, 32'h0, S ^ {16'h0, T[15:0]}  };
84
85              //other
86              INC     : begin
87                          Y_hi = 32'b0; {C, Y_lo} = S + 1;
88                          V = ~S[31] & Y_lo[31];//pos+1=neg
89                        end
90              INC4    : begin
91                          Y_hi = 32'b0; {C, Y_lo} = S + 4;
92                          V = ~S[31] & Y_lo[31];//pos+4=neg
93                        end
94              DEC     : begin
95                          Y_hi = 32'b0; {C, Y_lo} = S - 1;
96                          V =  S[31] &~Y_lo[31];//neg-1=pos
97                        end
98              DEC4    : begin
99                          Y_hi = 32'b0; {C, Y_lo} = S - 4;
100                         V =  S[31] &~Y_lo[31];//neg-1=pos
101                       end
102             ZEROS   : {V, C, Y_hi, Y_lo} = {2'bx, 64'h0};
103             ONES    : {V, C, Y_hi, Y_lo} = {2'bx, 32'h0, 32'hFFFFFFFF};
104             SP_INIT: {V, C, Y_hi, Y_lo} = {2'bx, 32'h0, 32'h3FC};
105             default: {V, C, Y_hi, Y_lo} = {66'h0};//set to all zeros for error
106         endcase
107
108 endmodule
```