

```

1  `timescale 1ns / 1ps
2  ///////////////////////////////////////////////////////////////////
3  // Company:
4  // Engineer:
5  //
6  // Create Date: 01.03.2016 09:07:25
7  // Design Name:
8  // Module Name: Sine_Cosine_CORDIC
9  // Project Name:
10 // Target Devices:
11 // Tool Versions:
12 // Description:
13 //
14 // Dependencies:
15 //
16 // Revision:
17 // Revision 0.01 - File Created
18 // Additional Comments:
19 //
20 ///////////////////////////////////////////////////////////////////
21
22
23 module Sine_Cosine_CORDIC #(parameter W=32, W_Exp=8, W_Sgf = 23, S_Exp = 9)//Single  ↗
Precision */ # (parameter W = 64, W_Exp = 11, W_Sgf = 52, S_Exp = 12) //-- Double  ↗
Precision */
24 (
25 //Input Signals
26 input wire clk, // Reloj del sistema.
27 input wire rst, // Señal de reset  ↗
del sistema.
28 input wire beg_fsm_cordic, // Señal de inicio  ↗
de la maquina de estados del módulo CORDIC.
29 input wire ack_cordic, // Señal de  ↗
acknowledge proveniente de otro módulo que indica que ha recibido el resultado del  ↗
modulo CORDIC.
30 input wire operation, // Señal que indica  ↗
si se realiza la operacion seno(1'b1) o coseno(1'b0).
31
32 input wire [W-1:0] data_in, // Dato de entrada,  ↗
contiene el angulo que se desea calcular en radianes.
33 input wire [1:0] shift_region_flag, // Señal que indica  ↗
si el ángulo a calcular esta fuera del rango de calculo del algoritmo CORDIC.
34
35 //Output Signals
36 output reg ready_cordic, // Señal de salida  ↗
que indica que se ha completado el calculo del seno/coseno.
37 output reg [W-1:0] data_output, // Bus de datos con  ↗
el valor final del angulo calculado.
38
39 output reg overflow_flag, // Bandera de  ↗
overflow de la operacion.
40 output reg underflow_flag // Bandera de  ↗
underflow de la operacion.

```

```

41
42 );
43
44 //Parametros
45 parameter r_mode = 2'b00;
46
47 //Declaracion de señales para instanciacion.
48 wire ready_add_subt; // Señal
proveniente del modulo de suma/resta que indica que ya se completo la operacion.
49 wire beg_add_subt; // Señal de salida
del modulo CORDIC que indica al modulo de suma/resta que inicie la operacion.
50 wire ack_add_subt; // Señal de salida
del modulo CORDIC que indica al modulo de suma/resta que el resultado se recibio
exitosamente.
51 wire op_add_subt; // Señal hacia el
módulo de suma/resta que indica si se va a realizar una suma(1'b0) o una resta(1'b1).
52 wire [W-1:0] result_add_subt; // Dato de entrada
del modulo CORDIC, contiene el resultado del módulo de suma/resta.
53 wire [W-1:0] add_subt_dataA; // Bus de datos
hacia el modulo de suma/resta con el valor al que se le desea aplicar dicha operacion.
54 wire [W-1:0] add_subt_dataB; // Bus de datos
hacia el modulo de suma/resta con el valor al que se le desea aplicar dicha operacion.
55
56
57 CORDIC_Coprocessor #(.W(W), .E(W_Exp), .M(W_Sgf)) cordic_coprocessor_sin_cos
58 (
59 .clk(clk), // Reloj del sistema.
60 .rst_cordic(rst), // Señal de reset
del sistema.
61 .beg_fsm_cordic(beg_fsm_cordic), // Señal de inicio
de la maquina de estados del módulo CORDIC.
62 .ack_cordic(ack_cordic), // Señal de
acknowledge proveniente de otro módulo que indica que ha recibido el resultado del
modulo CORDIC.
63 .operation(operation), // Señal que indica
si se realiza la operacion seno(1'b1) o coseno(1'b0).
64 .ready_add_subt(ready_add_subt), // Señal que indica
que se ha realizado la operacion de suma/resta en punto flotante.
65 .data_in(data_in), // Dato de entrada,
contiene el angulo que se desea calcular en radianes.
66 .shift_region_flag(shift_region_flag), // Señal que indica
si el ángulo a calcular esta fuera del rango de calculo del algoritmo CORDIC.
67 .result_add_subt(result_add_subt), // Dato de entrada,
contiene el resultado del módulo de suma/resta.
68 .ready_cordic(ready_cordic), // Señal de salida
que indica que se ha completado el calculo del seno/coseno.
69 .beg_add_subt(beg_add_subt), // Señal de salida
que indica que se debe de iniciar el modulo de suma/resta.
70 .ack_add_subt(ack_add_subt), // Señal que le
indica al modulo de suma/resta que se recibio el resultado de este modulo
correctamente.
71 .op_add_subt(op_add_subt), // Señal hacia el
módulo de suma/resta que indica si se va a realizar una suma(1'b0) o una resta(1'b1).

```

```

72  .add_subt_dataA(add_subt_dataA),           // Bus de datos      ↗
    hacia el modulo de suma/resta con el valor al que se le desea aplicar dicha operacion.
73  .add_subt_dataB(add_subt_dataB),           // Bus de datos      ↗
    hacia el modulo de suma/resta con el valor al que se le desea aplicar dicha operacion.
74  .data_output(data_output)                  // Bus de datos con   ↗
    el valor final del angulo calculado.
75  );
76
77
78  FPU_Add_Subtract_Function # (.W(W),.W_Exp(W_Exp),.W_Sgf(W_Sgf),.S_Exp(S_Exp))      ↗
    add_subt_module
79  (
80  .clk(clk),                                // Reloj del sistema.
81  .rst(rst),                                // Señal de reset     ↗
    del sistema.
82  .beg_FSM(beg_add_subt),                    // Señal de salida    ↗
    que indica que se debe de iniciar el modulo de suma/resta.
83  .rst_FSM(ack_add_subt),                    // Señal que le       ↗
    indica al modulo de suma/resta que se recibio el resultado de este modulo
    correctamente.
84  .Data_X(add_subt_dataA),                    // Bus de datos      ↗
    hacia el modulo de suma/resta con el valor al que se le desea aplicar dicha operacion.
85  .Data_Y(add_subt_dataB),                    // Bus de datos      ↗
    hacia el modulo de suma/resta con el valor al que se le desea aplicar dicha operacion.
86  .add_subt(op_add_subt),                    // Señal hacia el     ↗
    módulo de suma/resta que indica si se va a realizar una suma(1'b0) o una resta(1'b1).
87  .r_mode(r_mode),                          // Indica el modo     ↗
    de redondeo para el modulo de suma.
88  .overflow_flag(overflow_flag),              // Bandera de         ↗
    overflow de la operacion.
89  .underflow_flag(underflow_flag),            // Bandera de         ↗
    underflow de la operacion.
90  .ready(ready_add_subt),                    // Señal que indica   ↗
    que se ha realizado la operacion de suma/resta en punto flotante.
91  .final_result_ieee(result_add_subt)         // Dato de entrada,   ↗
    contiene el resultado del módulo de suma/resta.
92  );
93
94
95
96
97  endmodule
98

```