`timescale 1ns / 1ps

module HLSM\_MaxMin\_Struct(Clk, Rst, go, max\_diff, done);

input Clk, Rst, go;

output [7:0] max\_diff;

output done;

wire max\_ld, max\_clr, min\_clr, min\_ld, min\_sel;

wire i\_ld, i\_clr, maxdiff\_ld;

wire ai\_gt\_max, ai\_lt\_min, i\_lt\_256;

wire [7:0] a\_i;

wire [8:0] i;

////////////////////////////////////

**Controller** c1(Clk, Rst, go, ai\_gt\_max, ai\_lt\_min, i\_lt\_256,

done, max\_ld, max\_clr, min\_clr, min\_ld, min\_sel, i\_ld, i\_clr, maxdiff\_ld, R\_en);

////////////////////////////////////

//Register256\_8(R\_Addr, W\_Addr, R\_en, W\_en,R\_Data, W\_Data, Clk, Rst);

**Register256\_8** RF(i[7:0], 8'b0, R\_en, 1'b0 ,a\_i, 8'b0, Clk, Rst);

// module Register9bits (Clk, Rst, Ld, I, Q);

// module Comparator\_9bits (A, B, AltB, AeqB, AgtB);

endmodule

`timescale 1ns / 1ps

module **Controller**(Clk, Rst, go, ai\_gt\_max, ai\_lt\_min, i\_lt\_256,

done, max\_ld, max\_clr, min\_clr, min\_ld, min\_sel, i\_ld, i\_clr, maxdiff\_ld, R\_en);

**input Clk, Rst, go, ai\_gt\_max, ai\_lt\_min, i\_lt\_256;**

**output** reg done, max\_ld, max\_clr, min\_clr, min\_ld, min\_sel, i\_ld,

i\_clr, maxdiff\_ld, R\_en;

reg [3:0] state, statenext;

parameter sA = 0,sB = 1,sC = 2,sD = 3,sE = 4,sF = 5,sG = 6,sH = 7,sI = 8;

////////////////////////////////////

always @ (posedge Clk)begin

if(Rst == 1) state <= sA;

else state <= statenext;

end

////////////////////////////////////

always @ (**state, go, ai\_gt\_max, ai\_lt\_min, i\_lt\_256**) begin

max\_ld <= 0; max\_clr <= 0; min\_clr <= 0; min\_ld <= 0;

min\_sel <= 0; i\_ld <= 0; i\_clr <= 0; maxdiff\_ld <= 0; R\_en <= 0;

case(state)

sA: begin

if(go == 1) statenext <= sB;

else statenext <= sA;

end

sG: begin

**// complete code here**

statenext <= sH;

end

sH: begin

**// complete code here**

statenext <= sC;

end

sI: begin

**// complete code here**

statenext <= sA;

end

default: begin

statenext <= sA;

end

endcase

end

endmodule

sB: begin

**// complete code here**

statenext <= sC;

end

sC: begin

**if( )**

statenext <= sD;

else statenext <= sI;

end

sD: begin

**R\_en <= 1;**

**if( )**

statenext <= sE;

else statenext <= sF;

end

sE: begin

**// complete code here**

statenext <= sF;

end

sF: begin

R\_en <= 1;

**if( )**

statenext <= sG;

else statenext <= sH;

end