module pixel\_process (

//input

rst\_n\_scl,clk\_scl,

scl\_cfg\_flt,scl\_cfg\_mode,

scl\_i\_data\_r,scl\_i\_data\_en,

//output

o\_dff5,scl\_o\_data\_r);

//input

input rst\_n\_scl,clk\_scl;

input [7:0] scl\_i\_data\_r;

input scl\_cfg\_mode;

input [1:0] scl\_cfg\_flt;

input scl\_i\_data\_en;

//output

output [7:0] scl\_o\_data\_r;

output o\_dff5;

reg [7:0] scl\_o\_data\_r;

reg o\_dff0,o\_dff1,o\_dff2,o\_dff3,o\_dff4,o\_dff5;//output\_en DFF

reg signed [10:0] coef\_0,coef\_1,coef\_2,coef\_3;//filter

reg [7:0] in\_data\_r0,in\_data\_r1,in\_data\_r2,in\_data\_r3,in\_data\_r4,in\_data\_r5;//data\_in DFF

wire signed [8:0] re\_r0,re\_r1,re\_r2,re\_r3;//data\_in symbol

reg signed [20:0] dff\_r0,dff\_r1,dff\_r2,dff\_r3;//add and mul

reg signed [21:0] dff1\_r0,dff1\_r1;

reg signed [22:0] dff2\_r;

//filter

always @(posedge clk\_scl or negedge rst\_n\_scl) begin

if(!rst\_n\_scl) begin

coef\_0 <= 0;

coef\_1 <= 0;

coef\_2 <= 0;

coef\_3 <= 0;

end

else begin

case(scl\_cfg\_flt)

2'd0: begin

coef\_0 <= -3;

coef\_1 <= 498;

coef\_2 <= 18;

coef\_3 <= -1;

end

2'd1: begin

coef\_0 <= -38;

coef\_1 <= 376;

coef\_2 <= 202;

coef\_3 <= -28;

end

2'd2: begin

coef\_0 <= -28;

coef\_1 <= 202;

coef\_2 <= 376;

coef\_3 <= -38;

end

2'd3: begin

coef\_0 <= -1;

coef\_1 <= 18;

coef\_2 <= 498;

coef\_3 <= -3;

end

default: begin

coef\_0 <= 0;

coef\_1 <= 512;

coef\_2 <= 0;

coef\_3 <= 0;

end

endcase

end

end

////data\_in symbol

assign re\_r0 = {1'b0,in\_data\_r0};

assign re\_r1 = {1'b0,in\_data\_r1};

assign re\_r2 = {1'b0,in\_data\_r2};

assign re\_r3 = {1'b0,in\_data\_r3};

//data\_in DFF

always @(posedge clk\_scl or negedge rst\_n\_scl) begin

if(!rst\_n\_scl) begin

in\_data\_r0 <= 8'b0;

in\_data\_r1 <= 8'b0;

in\_data\_r2 <= 8'b0;

in\_data\_r3 <= 8'b0;

in\_data\_r4 <= 8'b0;

in\_data\_r5 <= 8'b0;

end

else begin

in\_data\_r0 <= scl\_i\_data\_r;

in\_data\_r1 <= in\_data\_r0;

in\_data\_r2 <= in\_data\_r1;

in\_data\_r3 <= in\_data\_r2;

in\_data\_r4 <= in\_data\_r3;

in\_data\_r5 <= in\_data\_r4;

end

end

//port processing

always @(posedge clk\_scl or negedge rst\_n\_scl) begin

if(!rst\_n\_scl) begin

dff\_r0 <= 0;

dff\_r1 <= 0;

dff\_r2 <= 0;

dff\_r3 <= 0;

end

else if(o\_dff2==1 && o\_dff3==0) begin//first

dff\_r0 <= re\_r0\*coef\_3;

dff\_r1 <= re\_r1\*coef\_2;

dff\_r2 <= re\_r2\*coef\_1;

dff\_r3 <= re\_r2\*coef\_0;

end

else if(o\_dff0==0 && o\_dff1==1) begin//dao shu 2

dff\_r0 <= re\_r1\*coef\_3;

dff\_r1 <= re\_r1\*coef\_2;

dff\_r2 <= re\_r2\*coef\_1;

dff\_r3 <= re\_r3\*coef\_0;

end

else if(o\_dff1==0 && o\_dff2==1) begin//last

dff\_r0 <= re\_r2\*coef\_3;

dff\_r1 <= re\_r2\*coef\_2;

dff\_r2 <= re\_r2\*coef\_1;

dff\_r3 <= re\_r3\*coef\_0;

end

else begin

dff\_r0 <= re\_r0\*coef\_3;

dff\_r1 <= re\_r1\*coef\_2;

dff\_r2 <= re\_r2\*coef\_1;

dff\_r3 <= re\_r3\*coef\_0;

end

end

always @(posedge clk\_scl or negedge rst\_n\_scl) begin

if(!rst\_n\_scl) begin

dff1\_r0 <= 0;

dff1\_r1 <= 0;

end

else begin

dff1\_r0 <= dff\_r0+dff\_r1;

dff1\_r1 <= dff\_r2+dff\_r3;

end

end

always @(posedge clk\_scl or negedge rst\_n\_scl) begin

if(!rst\_n\_scl) begin

dff2\_r <= 0;

end

else begin

dff2\_r <= dff1\_r0+dff1\_r1;

end

end

//平均

always @(posedge clk\_scl or negedge rst\_n\_scl) begin

if(!rst\_n\_scl) begin

scl\_o\_data\_r <= 8'b0;

end

else if(!scl\_cfg\_mode) begin

scl\_o\_data\_r <= in\_data\_r5;

end

else begin

if(dff2\_r<0) begin

scl\_o\_data\_r <= 8'b0;

end

else if(dff2\_r>130560) begin

scl\_o\_data\_r <= 255;

end

else begin

scl\_o\_data\_r <= dff2\_r >> 9;

end

end

end

//data\_out\_DFF

always @(posedge clk\_scl or negedge rst\_n\_scl) begin

if(!rst\_n\_scl) begin

o\_dff0 <= 0;

o\_dff1 <= 0;

o\_dff2 <= 0;

o\_dff3 <= 0;

o\_dff4 <= 0;

o\_dff5 <= 0;

end

else begin

o\_dff0 <= scl\_i\_data\_en;

o\_dff1 <= o\_dff0;

o\_dff2 <= o\_dff1;

o\_dff3 <= o\_dff2;

o\_dff4 <= o\_dff3;

o\_dff5 <= o\_dff4;

end

end

endmodule