module tb;

reg rst\_n\_scl;

reg clk\_scl;

reg scl\_i\_vsync;

reg scl\_i\_hsync;

reg scl\_i\_data\_en;

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

reg [7:0] scl\_i\_data\_g;

reg [7:0] scl\_i\_data\_b;

reg scl\_cfg\_mode;

reg scl\_cfg\_rsz;

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

wire scl\_o\_vsync;

wire scl\_o\_hsync;

wire scl\_o\_data\_en;

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

wire [7:0] scl\_o\_data\_g;

wire [7:0] scl\_o\_data\_b;

parameter h = 1280;

parameter v = 1024;

parameter size = h\*v\*3+54;

parameter en\_0 = 100;

reg[7:0] image\_in [0:size-1];

reg[7:0] image\_exp[0:size-1];

reg[7:0] image\_out[0:size-1];

integer fp\_r;

integer fp\_p;

integer fp\_w;

integer memory;

integer en\_0\_q;

integer q\_i;

integer q\_o;

reg[31:0] HEIGHT;

reg[31:0] WIDTH\_IN;

reg[31:0] WIDTH\_OUT;

top u3(

.rst\_n\_scl(rst\_n\_scl),

.clk\_scl(clk\_scl),

.scl\_i\_vsync(scl\_i\_vsync),

.scl\_i\_hsync(scl\_i\_hsync),

.scl\_i\_data\_en(scl\_i\_data\_en),

.scl\_i\_data\_r(scl\_i\_data\_r),

.scl\_i\_data\_g(scl\_i\_data\_g),

.scl\_i\_data\_b(scl\_i\_data\_b),

.scl\_cfg\_mode(scl\_cfg\_mode),

.scl\_cfg\_rsz(scl\_cfg\_rsz),

.scl\_cfg\_flt(scl\_cfg\_flt),

.scl\_o\_vsync(scl\_o\_vsync),

.scl\_o\_hsync(scl\_o\_hsync),

.scl\_o\_data\_en(scl\_o\_data\_en),

.scl\_o\_data\_r(scl\_o\_data\_r),

.scl\_o\_data\_g(scl\_o\_data\_g),

.scl\_o\_data\_b(scl\_o\_data\_b)

);

initial begin

rst\_n\_scl = 0;

clk\_scl = 0;

scl\_i\_vsync = 1;

scl\_i\_hsync = 1;

scl\_i\_data\_en = 0;

scl\_cfg\_mode = 0;

scl\_cfg\_rsz = 0;

scl\_cfg\_flt = 3;

#10 rst\_n\_scl = 1;

fp\_r=$fopen("input.bmp","rb");

memory=$fread(image\_in,fp\_r);

fp\_p=$fopen("exp.bmp","rb");

memory=$fread(image\_exp,fp\_p);

HEIGHT[7 :0 ]=image\_in[22];

HEIGHT[15:8 ]=image\_in[23];

HEIGHT[23:16]=image\_in[24];

HEIGHT[31:24]=image\_in[25];

WIDTH\_IN[7 :0 ]=image\_in[18];

WIDTH\_IN[15:8 ]=image\_in[19];

WIDTH\_IN[23:16]=image\_in[20];

WIDTH\_IN[31:24]=image\_in[21];

#500 q\_i=0;

while (q\_i<WIDTH\_IN\*HEIGHT\*3)

begin

@(posedge clk\_scl);

if(q\_i%(3\*WIDTH\_IN)==0 && en\_0\_q <= en\_0)

begin

en\_0\_q = en\_0\_q+1;

scl\_i\_data\_en = 0;

end

else

begin

en\_0\_q = 0;

scl\_i\_data\_en = 1;

scl\_i\_data\_r = image\_in[54+q\_i+0];

scl\_i\_data\_g = image\_in[54+q\_i+1];

scl\_i\_data\_b = image\_in[54+q\_i+2];

q\_i=q\_i+3;

end

end

#200 scl\_i\_data\_en = 0;

end

initial begin

#200 q\_o=0;

// head

for(q\_o=0;q\_o<54; q\_o=q\_o+1)

begin

image\_out[q\_o]=image\_in[q\_o];

end

WIDTH\_OUT[7 :0 ]=image\_out[18];

WIDTH\_OUT[15:8 ]=image\_out[19];

WIDTH\_OUT[23:16]=image\_out[20];

WIDTH\_OUT[31:24]=image\_out[21];

if(scl\_cfg\_mode == 0)

begin

WIDTH\_OUT = WIDTH\_OUT;

end

else if(scl\_cfg\_rsz == 0)

begin

WIDTH\_OUT = WIDTH\_OUT/2;

end

else begin

WIDTH\_OUT = WIDTH\_OUT/4;

end

image\_out[18]=WIDTH\_OUT[7 :0 ];

image\_out[19]=WIDTH\_OUT[15:8 ];

image\_out[20]=WIDTH\_OUT[23:16];

image\_out[21]=WIDTH\_OUT[31:24];

// image

q\_o=0;

while (q\_o<WIDTH\_OUT\*HEIGHT\*3) begin

@(posedge clk\_scl);

if(scl\_o\_data\_en) begin

image\_out[54+q\_o+0]=scl\_o\_data\_r;

image\_out[54+q\_o+1]=scl\_o\_data\_g;

image\_out[54+q\_o+2]=scl\_o\_data\_b;

if(image\_out[54+q\_o+0] != image\_exp[54+q\_o+0] ) begin

$display("ERROR" );

end

if(image\_out[54+q\_o+1] != image\_exp[54+q\_o+1] ) begin

$display("ERROR" );

end

if(image\_out[54+q\_o+2] != image\_exp[54+q\_o+2] ) begin

$display("ERROR" );

end

q\_o=q\_o+3;

end

end

// output

fp\_w=$fopen("out.bmp","wb");

for(q\_o=0;q\_o<WIDTH\_OUT\*HEIGHT\*3+54; q\_o=q\_o+1) begin

if(q\_o<54) begin

$fwrite(fp\_w, "%c", image\_exp[q\_o]);

end

else begin

$fwrite(fp\_w, "%c", image\_out[q\_o]);

end

end

$fclose(fp\_r);

$fclose(fp\_p);

$fclose(fp\_w);

$finish;

end

always #5 clk\_scl = !clk\_scl;

endmodule