**Low Pass Filter**

**MATLAB Function:**

function [x\_out, y\_out,r\_out,g\_out,b\_out]=hdllpf(x\_in, y\_in, r\_in, g\_in, b\_in)

persistent OrigImg

persistent x1 x2 y1 y2

D = [1/6 1/6 1/6;

1/6 1/6 1/6;

1/6 1/6 1/6];

RGB = [r\_in; g\_in; b\_in];

OrigImg\_1 = D\*RGB;

r\_out=OrigImg\_1(1);

g\_out=OrigImg\_1(2);

b\_out=OrigImg\_1(3);

x\_out = x2;

x2 = x1;

x1 = x\_in;

y\_out = y2;

y2 = y1;

y1 = y\_in;

end

**MATLAB Function Test Bench:**

WIDTH = 800;%u

HEIGHT = 450;%v

rout=zeros(450,800);

ImgData = double(imread("C:\Users\Public\lotus1.jpg"));

for y = 0:HEIGHT

for x = 0:WIDTH

if y >= 0 && y < HEIGHT && x >= 0 && x < WIDTH

b = ImgData(y+1,x+1,1);

g = ImgData(y+1,x+1,2);

r = ImgData(y+1,x+1,3);

else

b = 0;

g = 0;

r = 0;

end

[xOut, yOut,rout,gout,bout] =hdllpf (x, y, r, g, b);

Iout(y+1,x+1,1)=bout;

Iout(y+1,x+1,2)=gout;

Iout(y+1,x+1,3)=rout;

end

end

figure(1)

imshow(uint8(ImgData));

figure(2)

imshow(uint8(Iout),[]);

**Generated VHDL code:**

-- -------------------------------------------------------------

--

-- File Name: C:\Users\Public\codegen\hdllpf\hdlsrc\hdllpf\_fixpt.vhd

-- Created: 2024-04-27 15:35:15

--

-- Generated by MATLAB 9.12, MATLAB Coder 5.4 and HDL Coder 3.20

--

--

--

-- -------------------------------------------------------------

-- Rate and Clocking Details

-- -------------------------------------------------------------

-- Design base rate: 1

--

--

-- Clock Enable Sample Time

-- -------------------------------------------------------------

-- ce\_out 1

-- -------------------------------------------------------------

--

--

-- Output Signal Clock Enable Sample Time

-- -------------------------------------------------------------

-- x\_out ce\_out 1

-- y\_out ce\_out 1

-- r\_out ce\_out 1

-- g\_out ce\_out 1

-- b\_out ce\_out 1

-- -------------------------------------------------------------

--

-- -------------------------------------------------------------

-- -------------------------------------------------------------

--

-- Module: hdllpf\_fixpt

-- Source Path: hdllpf\_fixpt

-- Hierarchy Level: 0

--

-- -------------------------------------------------------------

LIBRARY IEEE;

USE IEEE.std\_logic\_1164.ALL;

USE IEEE.numeric\_std.ALL;

USE work.hdllpf\_fixpt\_pkg.ALL;

ENTITY hdllpf\_fixpt IS

PORT( clk : IN std\_logic;

reset : IN std\_logic;

clk\_enable : IN std\_logic;

x\_in : IN std\_logic\_vector(9 DOWNTO 0); -- ufix10

y\_in : IN std\_logic\_vector(8 DOWNTO 0); -- ufix9

r\_in : IN std\_logic\_vector(7 DOWNTO 0); -- ufix8

g\_in : IN std\_logic\_vector(7 DOWNTO 0); -- ufix8

b\_in : IN std\_logic\_vector(7 DOWNTO 0); -- ufix8

ce\_out : OUT std\_logic;

x\_out : OUT std\_logic\_vector(9 DOWNTO 0); -- ufix10

y\_out : OUT std\_logic\_vector(8 DOWNTO 0); -- ufix9

r\_out : OUT std\_logic\_vector(6 DOWNTO 0); -- ufix7

g\_out : OUT std\_logic\_vector(6 DOWNTO 0); -- ufix7

b\_out : OUT std\_logic\_vector(6 DOWNTO 0) -- ufix7

);

END hdllpf\_fixpt;

ARCHITECTURE rtl OF hdllpf\_fixpt IS

-- Signals

SIGNAL enb : std\_logic;

SIGNAL x\_in\_unsigned : unsigned(9 DOWNTO 0); -- ufix10

SIGNAL x2\_reg\_reg : vector\_of\_unsigned10(0 TO 1); -- ufix10 [2]

SIGNAL x2 : unsigned(9 DOWNTO 0); -- ufix10

SIGNAL y\_in\_unsigned : unsigned(8 DOWNTO 0); -- ufix9

SIGNAL y2\_reg\_reg : vector\_of\_unsigned9(0 TO 1); -- ufix9 [2]

SIGNAL y2 : unsigned(8 DOWNTO 0); -- ufix9

SIGNAL r\_out\_tmp : unsigned(6 DOWNTO 0); -- ufix7

SIGNAL g\_out\_tmp : unsigned(6 DOWNTO 0); -- ufix7

SIGNAL b\_out\_tmp : unsigned(6 DOWNTO 0); -- ufix7

BEGIN

x\_in\_unsigned <= unsigned(x\_in);

enb <= clk\_enable;

x2\_reg\_process : PROCESS (clk, reset)

BEGIN

IF reset = '1' THEN

x2\_reg\_reg <= (OTHERS => to\_unsigned(16#000#, 10));

ELSIF clk'EVENT AND clk = '1' THEN

IF enb = '1' THEN

x2\_reg\_reg(0) <= x\_in\_unsigned;

x2\_reg\_reg(1) <= x2\_reg\_reg(0);

END IF;

END IF;

END PROCESS x2\_reg\_process;

x2 <= x2\_reg\_reg(1);

x\_out <= std\_logic\_vector(x2);

y\_in\_unsigned <= unsigned(y\_in);

y2\_reg\_process : PROCESS (clk, reset)

BEGIN

IF reset = '1' THEN

y2\_reg\_reg <= (OTHERS => to\_unsigned(16#000#, 9));

ELSIF clk'EVENT AND clk = '1' THEN

IF enb = '1' THEN

y2\_reg\_reg(0) <= y\_in\_unsigned;

y2\_reg\_reg(1) <= y2\_reg\_reg(0);

END IF;

END IF;

END PROCESS y2\_reg\_process;

y2 <= y2\_reg\_reg(1);

y\_out <= std\_logic\_vector(y2);

-- HDL code generation from MATLAB function: hdllpf\_fixpt

--

-- %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

--

-- %

--

-- Generated by MATLAB 9.12 and Fixed-Point Designer 7.4 %

--

-- %

--

-- %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

r\_out\_tmp <= to\_unsigned(16#00#, 7);

r\_out <= std\_logic\_vector(r\_out\_tmp);

g\_out\_tmp <= to\_unsigned(16#00#, 7);

g\_out <= std\_logic\_vector(g\_out\_tmp);

b\_out\_tmp <= to\_unsigned(16#00#, 7);

b\_out <= std\_logic\_vector(b\_out\_tmp);

ce\_out <= clk\_enable;

END rtl;

**Generated Verilog Code:**

// -------------------------------------------------------------

//

// File Name: C:\Users\Public\codegen\hdllpf\hdlsrc\hdllpf\_fixpt.v

// Created: 2024-04-27 15:33:43

//

// Generated by MATLAB 9.12, MATLAB Coder 5.4 and HDL Coder 3.20

//

//

//

// -- -------------------------------------------------------------

// -- Rate and Clocking Details

// -- -------------------------------------------------------------

// Design base rate: 1

//

//

// Clock Enable Sample Time

// -- -------------------------------------------------------------

// ce\_out 1

// -- -------------------------------------------------------------

//

//

// Output Signal Clock Enable Sample Time

// -- -------------------------------------------------------------

// x\_out ce\_out 1

// y\_out ce\_out 1

// r\_out ce\_out 1

// g\_out ce\_out 1

// b\_out ce\_out 1

// -- -------------------------------------------------------------

//

// -------------------------------------------------------------

// -------------------------------------------------------------

//

// Module: hdllpf\_fixpt

// Source Path: hdllpf\_fixpt

// Hierarchy Level: 0

//

// -------------------------------------------------------------

`timescale 1 ns / 1 ns

module hdllpf\_fixpt

(clk,

reset,

clk\_enable,

x\_in,

y\_in,

r\_in,

g\_in,

b\_in,

ce\_out,

x\_out,

y\_out,

r\_out,

g\_out,

b\_out);

input clk;

input reset;

input clk\_enable;

input [9:0] x\_in; // ufix10

input [8:0] y\_in; // ufix9

input [7:0] r\_in; // ufix8

input [7:0] g\_in; // ufix8

input [7:0] b\_in; // ufix8

output ce\_out;

output [9:0] x\_out; // ufix10

output [8:0] y\_out; // ufix9

output [6:0] r\_out; // ufix7

output [6:0] g\_out; // ufix7

output [6:0] b\_out; // ufix7

wire enb;

reg [9:0] x2\_reg\_reg [0:1]; // ufix10 [2]

wire [9:0] x2\_reg\_reg\_next [0:1]; // ufix10 [2]

wire [9:0] x2; // ufix10

reg [8:0] y2\_reg\_reg [0:1]; // ufix9 [2]

wire [8:0] y2\_reg\_reg\_next [0:1]; // ufix9 [2]

wire [8:0] y2; // ufix9

assign enb = clk\_enable;

always @(posedge clk or posedge reset)

begin : x2\_reg\_process

if (reset == 1'b1) begin

x2\_reg\_reg[0] <= 10'b0000000000;

x2\_reg\_reg[1] <= 10'b0000000000;

end

else begin

if (enb) begin

x2\_reg\_reg[0] <= x2\_reg\_reg\_next[0];

x2\_reg\_reg[1] <= x2\_reg\_reg\_next[1];

end

end

end

assign x2 = x2\_reg\_reg[1];

assign x2\_reg\_reg\_next[0] = x\_in;

assign x2\_reg\_reg\_next[1] = x2\_reg\_reg[0];

always @(posedge clk or posedge reset)

begin : y2\_reg\_process

if (reset == 1'b1) begin

y2\_reg\_reg[0] <= 9'b000000000;

y2\_reg\_reg[1] <= 9'b000000000;

end

else begin

if (enb) begin

y2\_reg\_reg[0] <= y2\_reg\_reg\_next[0];

y2\_reg\_reg[1] <= y2\_reg\_reg\_next[1];

end

end

end

assign y2 = y2\_reg\_reg[1];

assign y2\_reg\_reg\_next[0] = y\_in;

assign y2\_reg\_reg\_next[1] = y2\_reg\_reg[0];

// HDL code generation from MATLAB function: hdllpf\_fixpt

//

// %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

//

// %

//

// Generated by MATLAB 9.12 and Fixed-Point Designer 7.4 %

//

// %

//

// %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

assign r\_out = 7'b0000000;

assign g\_out = 7'b0000000;

assign b\_out = 7'b0000000;

assign ce\_out = clk\_enable;

assign x\_out = x2;

assign y\_out = y2;

endmodule // hdllpf\_fixpt

**Package:**

-- -------------------------------------------------------------

--

-- File Name: C:\Users\Public\codegen\hdllpf\hdlsrc\hdllpf\_fixpt\_pkg.vhd

-- Created: 2024-04-27 15:35:15

--

-- Generated by MATLAB 9.12, MATLAB Coder 5.4 and HDL Coder 3.20

--

--

-- -------------------------------------------------------------

LIBRARY IEEE;

USE IEEE.std\_logic\_1164.ALL;

USE IEEE.numeric\_std.ALL;

PACKAGE hdllpf\_fixpt\_pkg IS

TYPE vector\_of\_unsigned10 IS ARRAY (NATURAL RANGE <>) OF unsigned(9 DOWNTO 0);

TYPE vector\_of\_unsigned9 IS ARRAY (NATURAL RANGE <>) OF unsigned(8 DOWNTO 0);

END hdllpf\_fixpt\_pkg;

**Input Image:**



**Output Image:**

