EE 310 Hardware Description Languages
Spring 2022

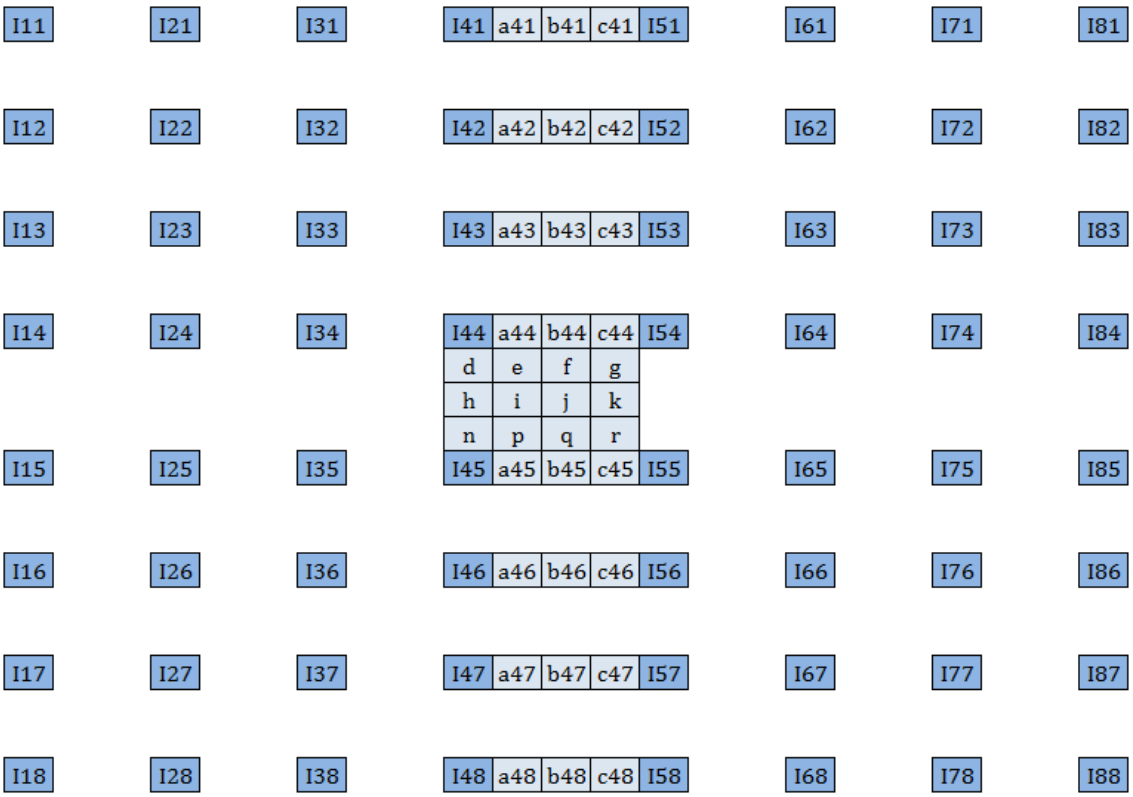
Laboratory Assignment #2

Due Date: 18 April 2022

Design a sequential hardware which implements the following sub-pixel interpolation algorithm.

An 8x8 block containing integer pixels I11 to I88 is shown below. The sub-pixel interpolation algorithm interpolates half-pixels a, b, c and d, h, n from integer pixels in horizontal and vertical direction, respectively, using three different Finite Impulse Response (FIR) filters. Half-pixels a and d are interpolated using type A filter with weights (1/64, 2/64, 6/64, 46/64, 6/64, 2/64, 1/64). Half-pixels b and h are interpolated using type B filter with weights (1/64, 4/64, 6/64, 42/64, 6/64, 4/64, 1/64). Half-pixels c and n are interpolated using type C filter with weights (1/64, 10/64, 4/64, 34/64, 4/64, 10/64, 1/64). For example, half-pixels **a41**, **b41**, **c41** in the figure given below are interpolated as follows:

a41 = (I11 + 2*I21 + 6*I31 + 46*I41 + 6*I51 + 2*I61 + I71) / 64
b41 = (I11 + 4*I21 + 6*I31 + 42*I41 + 6*I51 + 4*I61 + I71) / 64
c41 = (I21 + 10*I31 + 4*I41 + 34*I51 + 4*I61 + 10*I71 + I81) / 64



After all half-pixels a, b, c are interpolated, the quarter-pixels e, f, g are interpolated from half-pixels a, b, c respectively in vertical direction using type A filter, the quarter-pixels i, j, k are interpolated similarly using type B filter, and the quarter-pixels p, q, r are interpolated similarly using type C filter. For example, quarter-pixels **e**, **j**, **r** in the figure given above are interpolated as follows:

e = (a41 + 2*a42 + 6*a43 + 46*a44 + 6*a45 + 2*a46 + a47) / 64
j = (b41 + 4*b42 + 6*b43 + 42*b44 + 6*b45 + 4*b46 + b47) / 64
r = (c42 + 10*c43 + 4*c44 + 34*c45 + 4*c46 + 10*c47 + c48) / 64

I11 to I88 values (integers in the range [0-255]) will be given as inputs to the interpolation hardware. The hardware should start working after the input signal *start* is high. It should produce the output values a44, b44, c44, d, h, n, e, i, p, f, j, q, g, k, r (integers in the range [0-255]) by implementing the FIR filter equations. Then, it should set the output signal *done* to high. Assume that the I11 to I88 inputs will contain the I11 to I88 values until the outputs are produced.

In your hardware, you are not allowed to use multiplier or divider hardware for implementing multiplication with a constant and division by constant operations. These operations should be implemented using shifter and adder hardware.

Your hardware should interpolate only 3 sub-pixels in a clock cycle. In your hardware, you are allowed to use as many shifter and adder hardware as necessary for interpolating 3 sub-pixels in parallel.

Implement your hardware design using Verilog HDL. Your top-level Verilog module should have the following interface:

module sp_interpolator (clk, reset, start, I11,I12,I13,I14,I15,I16,I17,I18,I21,I22,
I23,I24,I25,I26,I27,I28,I31,I32,I33,I34,I35,I36,I37,I38,I41,I42,I43,I44,I45,I46,I47,
I48,I51,I52,I53,I54,I55,I56,I57,I58,I61,I62,I63,I64,I65,I66,I67,I68,I71,I72,I73,I74,I75,
I76,I77,I78,I81,I82,I83, I84,I85,I86,I87,I88,*a44,b44,c44,d,h,n,e,i,p,f,j,q,g,k,r,done*);
input clk, reset, start;
input [7:0] I11,I12,I13,I14,I15,I16,I17,I18,I21,I22,I23,I24,I25,I26,I27,I28,
I31,I32,I33,I34,I35,I36,I37,I38,I41,I42,I43,I44,I45,I46,I47,I48,
I51,I52,I53,I54,I55,I56,I57,I58,I61,I62,I63,I64,I65,I66,I67,I68,
I71,I72,I73,I74,I75,I76,I77,I78,I81,I82,I83,I84,I85,I86,I87,I88;
output [7:0] a44,b44,c44,d,h,n,e,i,p,f,j,q,g,k,r;
output done;
…..
endmodule

Write a Verilog testbench that verifies the correctness of your Verilog implementation. In this testbench, you should apply input values to your Verilog implementation and compare its outputs with the expected results.

Put all your Verilog files into a zip file named Lab2_Partner1Lastname_ Partner2Lastname.zip (e.g., *Lab2_Mahdavi_Hamzaoglu.zip*) and submit this zip file to EE310 SUCourse+.