HLS ASSIGNMENT-7(KANEKAL KOUSAR)

(Q) Model the FIR Filter design in the most efficient manner possible for the hardware (e.g. small clock period, lower initiation interval, less resource consumption, etc.). After designing, you should compare your output against the reference output generated by the C code for the same set of input vectors, using a self checking testbench that allows for a 5% difference in output values generated by the C code and the HLS code. Use at least two different input vectors. Also, the HLS design should use appropriate fixed point format instead of floating point format wherever applicable. The C code from the website can be integrated as part of your HLS testbench if you name both the design modules differently, for e.g., firFloat for C module and firFixed for HLS module. This will enable you to pass the input vectors to both the modules and compare the outputs, in a single testbench.

HEADER FILE

```
#include <ap_fixed.h>
#include <ap_int.h>

typedef ap_fixed<24,16> int16_dt;
typedef ap_fixed<40,32> int32_dt;


#include<hls_stream.h>
using namespace hls;
// maximum number of inputs that can be handled in one function call
#define MAX_INPUT_LEN   10
// maximum length of filter than can be handled
#define MAX_FLT_LEN     10
// buffer to hold all of the input samples
#define BUFFER_LEN      (MAX_FLT_LEN - 1 + MAX_INPUT_LEN)


//for fixed points
struct input{
       int16_dt in[MAX_INPUT_LEN];
};

struct coeff{
       int16_dt cof[MAX_FLT_LEN];
};

struct output{
       int16_dt out[MAX_INPUT_LEN];
};

//for float points
struct inputf{
       double in[MAX_INPUT_LEN];
};
```

```
struct coefff{
       double cof[MAX_FLT_LEN];
};

struct outputf{
       double out[MAX_INPUT_LEN];
};
```

DESIGN CODE

```
#include "header.h"
#include<ap_int.h>

void fir_fix( stream<coeff> &coeffs, stream<input> &inputs,
stream<output> &outputs,ap_int<4> length, ap_int<4> filterLength )
{
    int32_dt acc;     //accumulator
    input data=inputs.read();
//stores input data
    coeff c=coeffs.read();
//stores filter coefficients
    output out;

    //initialize the buffer to zero
    int16_dt insamp[ BUFFER_LEN ]={0};
#pragma HLS ARRAY_RESHAPE variable=insamp block factor=10 dim=1
   /* for (int i=0;i<BUFFER_LEN;i++){
#pragma HLS PIPELINE
#pragma HLS LOOP_TRIPCOUNT
       insamp[i]=0;
    }*/

  //   put the new samples at the high end of the buffer
       for (int i=0;i<length;i++){
#pragma HLS PIPELINE
#pragma HLS LOOP_TRIPCOUNT
       insamp[filterLength-1+i]=data.in[i];
    }


    // apply the filter to each input sample
    for ( int n = 0; n <filterLength-1+length; n++ ){
#pragma HLS UNROLL
#pragma HLS LOOP_TRIPCOUNT
        // calculate output n
       ap_int<8> co=0;
       ap_int<16> inp=filterLength-1+n;
         acc = 0;
        for (int k = 0; k < filterLength; k++ )
#pragma HLS PIPELINE
 {
#pragma HLS LOOP_TRIPCOUNT
            acc += (c.cof[co]) * (insamp[inp]);
            co++;
            inp--;
```

```
        }
        out.out[n]= acc;
        }

    outputs<<out;
}



void fir_float( stream<coefff> &coeffs, stream<inputf> &inputs,
stream<outputf> &outputs,int length, int filterLength )
{
    double acc;     // accumulator for MACs
    inputf data=inputs.read();
    coefff c=coeffs.read();
    outputf o;

    double insamp[ BUFFER_LEN ];
    for (int i=0;i<BUFFER_LEN;i++){
       insamp[i]=0;
    }

    // put the new samples at the high end of the buffer
    for (int i=0;i<length;i++){
       insamp[filterLength-1+i]=data.in[i];
    }


    // apply the filter to each input sample
    for ( int n = 0; n <filterLength-1+length; n++ ) {
        // calculate output n
       int co=0;
       int inp=filterLength-1+n;
        acc = 0;
        for (int k = 0; k < filterLength; k++ ) {
            acc += (c.cof[co]) * (insamp[inp]);
            co++;
            inp--;
        }
        o.out[n]= acc;
        }

    // shift input samples back in time for next time
    for (int i=0;i<length;i++){
             insamp[i]=insamp[filterLength+i];
        }

    outputs<<o;
}
```

TEST BENCH CODE

```
#include "header.h"
#include<iostream>
using namespace std;

#include <fstream>

void fir_fix(stream<coeff> &coeffs, stream<input> &inputs,
stream<output> &outputs,ap_int<4> length, ap_int<4> filterLength );
void fir_float( stream<coefff> &coeffs, stream<inputf> &inputs,
stream<outputf> &outputs,int length, int filterLength );

int main(){

	int length=7;
	int filterLength=6;

	stream<coeff> coeffs;
	coeff c={1,0,-1,0,1,0,-1};
	coeffs<<c;

	stream<coefff> coeffs2;
	coefff c2={1,0,-1,0,1,0,-1};
	coeffs2<<c2;

	//fixed point format
	stream<input> inputs;
	input i={0.1,0.2,0.4,0.2,0.1};
	inputs<<i;
	stream<output> outputs;

	fir_fix(coeffs,inputs,outputs,length,filterLength);
	output o1=outputs.read();
	cout<<endl;
	cout<<"hls code ouput:";
	for (int i=0;i<filterLength-1+length;i++){
		cout<<o1.out[i]<<", ";
	}
	cout<<endl;
	//for floating points
	stream<inputf> inputs2;
	inputf i2={0.1,0.2,0.4,0.2,0.1};
	inputs2<<i2;
	stream<outputf> outputs2;

	fir_float(coeffs2,inputs2,outputs2,length,filterLength);
	outputf o2=outputs2.read();
	cout<<"c code ouput:";
	for (int i=0;i<filterLength-1+length;i++){
		cout<<o2.out[i]<<", ";
		}
}
```

Synthesis report:

Synthesis Report for 'fir_fix'

General Information

Date:	Wed Apr 5 20:33:35 2023
Version:	2017.4 (Build 2086221 on Fri Dec 15 21:13:33 MST 2017)
Project:	assignment7
Solution:	solution1
Product family:	zynq
Target device:	xc7z020clg484-1

Performance Estimates

Timing (ns)

Summary

Clock	Target	Estimated	Uncertainty
ap_clk	10.00	5.96	1.25

Latency (clock cycles)

Summary

Latency		Interval		
min	max	min	max	Type
6	6	6	6	none

Detail

Instance

Loop

Utilization Estimates

Summary

Name	BRAM_18K	DSP48E	FF	LUT
DSP	-	-	-	-
Expression	-	-	0	86
FIFO	-	-	-	-
Instance	-	-	-	-
Memory	0	-	2	1
Multiplexer	-	-	-	391
Register	-	-	25	-
Total	0	0	27	478
Available	280	220	106400	53200
Utilization (%)	0	0	~0	~0

Detail

Instance

DSP48

Memory

FIFO

Expression

Multiplexer

Simulation:

```
INFO: [SIM 2] *************** CSIM start ***************
INFO: [SIM 4] CSIM will launch GCC as the compiler.
   Compiling ../../../fir_fil_tb.cpp in debug mode
   Generating csim.exe

hls code ouput:0.0976563, 0.199219, 0.300781, 0, -0.203125, 0, 0.300781,
0.199219, 0.0976563, 0, 0, 0,
c code ouput:0.1, 0.2, 0.3, 0, -0.2, 0, 0.3, 0.2, 0.1, 0, 0, 0,
INFO: [SIM 1] CSim done with 0 errors.
INFO: [SIM 3] *************** CSIM finish ***************
```