Assignment-8(kanekal kousar)

The purpose of this assignment is to familiarise you with the Vivado and HLS flow. General Instructions:
1. Read about Cyclic Prefix removal in 5G NR from the PDF (pg. 11) attached with this mail.

2. Write an HLS module, along with testbench, for Cyclic Prefix removal from an incoming set of data inputs.

3. After HLS testing using HLS testbench and synthesis of the module in HLS, export the module to Vivado application.

4. The input to the CP removal module should be given from another HLS module which will act as a data generator. Use an array to store the input values from the attached data set file in the mail and perform synthesis in HLS to get the data generator module.

5. Interface both the modules with Xilinx FFT IP and perform FFT on the output of Cyclic Prefix Removal in the Vivado application.

6. Generate a wrapper for the block diagram in Vivado and then write a test bench in Verilog which will provide clock and reset to these modules and store the FFT output in a file.

7. Run simulation.

8. Compare the output obtained from Verilog by writing a code in MATLAB for the same functionality and check for correctness.

Generator code :

```
#include <complex>
using namespace std;
typedef complex<float> cmp;
#include <hls_stream.h>
using namespace hls;

void generator(stream<cmp> &out){
#pragma HLS INTERFACE axis register both port=out

float r[]={0.364920, -0.752842,……}

float img[]={-0.728851, 0.31251……..}

for (int i=0;i<8800;i++){
#pragma HLS PIPELINE II=1
       out.write(complex<float>(r[i],img[i]));
       }
}
```

Desing code:

```
#include <complex>
using namespace std;
typedef complex<float> cmp;
#include <hls_stream.h>
using namespace hls;

void cp(stream<cmp> &in,stream<cmp> &out,bool &tlast){
#pragma HLS INTERFACE ap_none port=tlast
#pragma HLS DATA_PACK variable=out
#pragma HLS DATA_PACK variable=in
#pragma HLS INTERFACE axis off port=out
#pragma HLS INTERFACE axis off port=in

	cmp ind;
	for (int z=1;z<=8800;z++){
#pragma HLS PIPELINE II=1
		ind=in.read();
	if (z>320 && z<= 4416 || (z>4704 && z<= 8800)){
		out.write(ind);
	}
	if (z==8800){
		tlast=true;
	}
	else{
		tlast=false;
	}
	}
}
```

Test bench code:

```
#include <hls_stream.h>
#include <complex>
#include <iostream>
#include <fstream>
using namespace hls;
using namespace std;
typedef complex<float> cmp;
void generator(stream<cmp> &out);
void cp(stream<cmp> &in,stream<cmp> &out,bool &last);

int main(){
	stream<cmp> out;
	stream<cmp> cp_out;
	generator(out);
	cmp o;
	bool last;
	/*
	ofstream gen_output("gen_output.dat");
	for (int i=0;i<8800;i++){
		o=out.read();
		gen_output<<o<<endl;//storing the generator output
```

```
        }*/
        cp(out,cp_out,last);
        ofstream output("cp_output.dat");
                for (int i=0;i<8192;i++){
                        o=cp_out.read();
                        output<<o<<endl;//storing cp remover output
                }
                // Compare the results file with the ref file
                                int retval = system("diff --brief -w cp_output.dat ref.dat");
                                if (retval != 0) {
                                        printf("Test failed  !!!\n");
                                        retval=1;
                                } else {
                                        printf("Test passed !\n");
}}
```

FILES

Input to cp_remover:https://github.com/kkousar/HLS/blob/main/hls%20files/gen_output.dat

Output of cp_remover: https://github.com/kkousar/HLS/blob/main/hls%20files/cp_output.dat

Ref input to testbench:https://github.com/kkousar/HLS/blob/main/hls%20files/ref.dat

Generator Synthesis report:

Synthesis Report for 'generator'

General Information

Date:	Wed May 24 11:46:10 2023
Version:	2017.4 (Build 2086221 on Fri Dec 15 21:13:33 MST 2017)
Project:	cp_remover2
Solution:	solution1
Product family:	zynq
Target device:	xc7z020clg484-1

Performance Estimates

Timing (ns)

Summary

Clock	Target	Estimated	Uncertainty
ap_clk	10.00	3.25	1.25

Latency (clock cycles)

Summary

Latency		Interval		
min	max	min	max	Type
8802	8802	8802	8802	none

Detail

Instance

Loop

Utilization Estimates

Summary

Name	BRAM_18K	DSP48E	FF	LUT
DSP	-	-	-	-
Expression	-	-	0	74
FIFO	-	-	-	-
Instance	-	-	-	-

Instance	-	-	-	-
Memory	64	-	0	0
Multiplexer	-	-	-	63
Register	-	-	21	-
Total	64	0	21	137
Available	280	220	106400	53200
Utilization (%)	22	0	~0	~0

Detail

- Instance
- DSP48
- Memory
- FIFO
- Expression
- Multiplexer
- Register

Interface

Summary

RTL Ports	Dir	Bits	Protocol	Source Object	C Type
ap_clk	in	1	ap_ctrl_hs	generator	return value
ap_rst_n	in	1	ap_ctrl_hs	generator	return value
ap_start	in	1	ap_ctrl_hs	generator	return value
ap_done	out	1	ap_ctrl_hs	generator	return value
ap_idle	out	1	ap_ctrl_hs	generator	return value
ap_ready	out	1	ap_ctrl_hs	generator	return value
out_V_TDATA	out	64	axis	out_V	pointer
out_V_TVALID	out	1	axis	out_V	pointer
out_V_TREADY	in	1	axis	out_V	pointer

Cp_remover synthesis :

Synthesis Report for 'cp'

General Information

Date:	Fri May 26 21:31:59 2023
Version:	2017.4 (Build 2086221 on Fri Dec 15 21:13:33 MST 2017)
Project:	cp_remover2
Solution:	solution2
Product family:	zynq
Target device:	xc7z020clg484-1

Performance Estimates

Timing (ns)

Summary

Clock	Target	Estimated	Uncertainty
ap_clk	10.00	3.14	1.25

Latency (clock cycles)

Summary

Latency		Interval		
min	max	min	max	Type
8802	8802	8802	8802	none

Detail

- Instance
- Loop

Utilization Estimates

Summary

Name	BRAM_18K	DSP48E	FF	LUT
DSP	-	-	-	-
Expression	-	-	0	158
FIFO	-	-	-	-
Instance	-	-	-	-

Memory	-	-	-	-
Multiplexer	-	-	-	81
Register	-	-	86	-
Total	0	0	86	239
Available	280	220	106400	53200
Utilization (%)	0	0	~0	~0

Detail

- Instance
- DSP48
- Memory
- FIFO
- Expression
- Multiplexer
- Register

Interface

Summary

RTL Ports	Dir	Bits	Protocol	Source Object	C Type
ap_clk	in	1	ap_ctrl_hs	cp	return value
ap_rst_n	in	1	ap_ctrl_hs	cp	return value
ap_start	in	1	ap_ctrl_hs	cp	return value
ap_done	out	1	ap_ctrl_hs	cp	return value
ap_idle	out	1	ap_ctrl_hs	cp	return value
ap_ready	out	1	ap_ctrl_hs	cp	return value
in_V_TDATA	in	64	axis	in_V	pointer
in_V_TVALID	in	1	axis	in_V	pointer
in_V_TREADY	out	1	axis	in_V	pointer
out_V_TDATA	out	64	axis	out_V	pointer
out_V_TVALID	out	1	axis	out_V	pointer
out_V_TREADY	in	1	axis	out_V	pointer
tlast	out	1	ap_none	tlast	pointer

Simulation report

```
INFO: [SIM 2] *************** CSIM start ***************
INFO: [SIM 4] CSIM will launch GCC as the compiler.
make: `csim.exe' is up to date.
Test passed !
INFO: [SIM 1] CSim done with 0 errors.
INFO: [SIM 3] *************** CSIM finish ***************
```

Co-simulation report:

```
INFO: [Common 17-206] Exiting xsim at Fri May 26 21:37:14 2023...
INFO: [COSIM 212-316] Starting C post checking ...
Test passed !
INFO: [COSIM 212-1000] *** C/RTL co-simulation finished: PASS ***
INFO: [COSIM 212-211] II is measurable only when transaction number is greater than 1
in RTL simulation. Otherwise, they will be marked as all NA. If user wants to
calculate them, please make sure there are at least 2 transactions in RTL simulation.
Finished C/RTL cosimulation.
```

Cosimulation Report for 'cp'

Result

		Latency			Interval		
RTL	Status	min	avg	max	min	avg	max
VHDL	NA	NA	NA	NA	NA	NA	NA
Verilog	Pass	8802	8802	8802	NA	NA	NA

Export the report(.html) using the Export Wizard

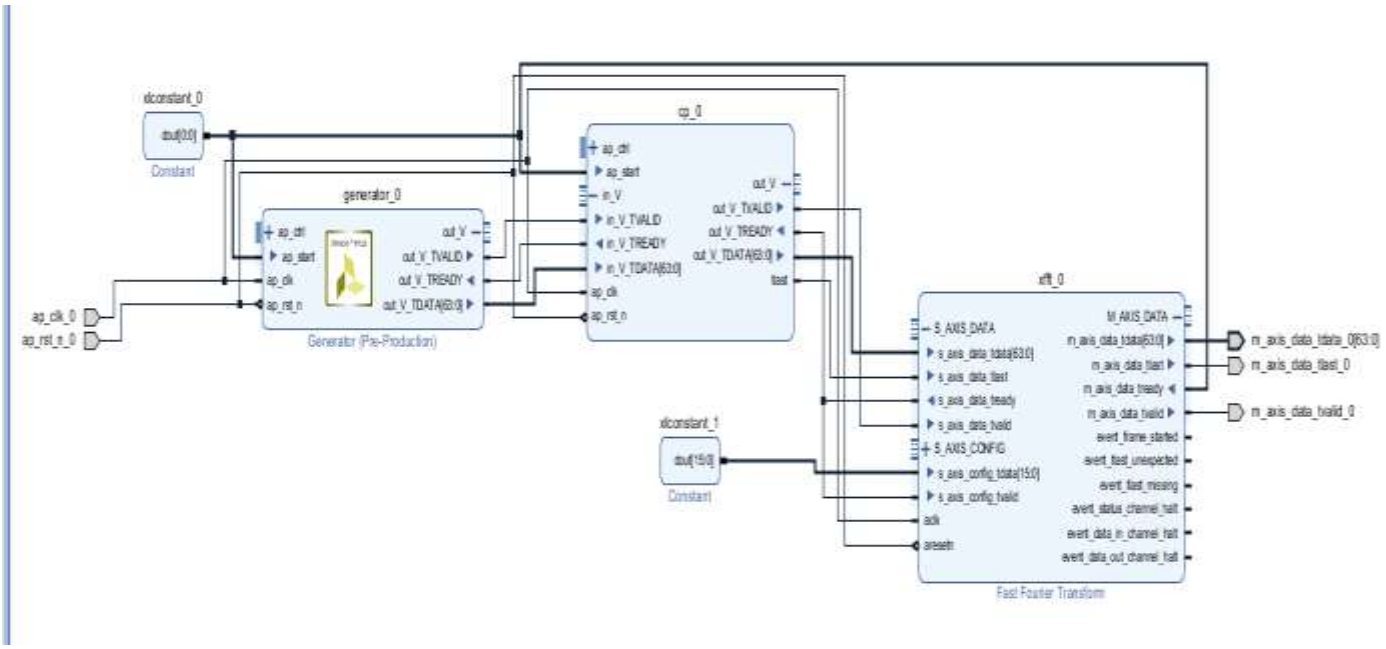
Matlab code:

```
file1 = fopen('/MATLAB Drive/puschTxAfterChannelImag.txt','r')
file2 = fopen('/MATLAB Drive/puschTxAfterChannelReal.txt','r')
file3=fopen('output_cp.txt','w')
file4=fopen('output.txt','w')

cp_added=[]
cp_removed=[]
c=0

while ~feof(file1) && ~feof(file2)
    line1 = fgetl(file1);
    line2 = fgetl(file2);
    cp_added = [cp_added; complex(str2double(line2), str2double(line1))];
end



for z = 1:8800
    if (z > 320 && z <= 4416) || (z > 4704 && z <= 8800)
        cp_removed = [cp_removed; cp_added(z)];
    end
end
for k = 1:numel(cp_removed)
    fprintf(file4, '%f +i%f\n', real(cp_removed(k)), imag(cp_removed(k)));
end

fft_result = fft(cp_removed);
for k = 1:numel(fft_result)
    fprintf(file3, '%f +i%f\n', real(fft_result(k)), imag(fft_result(k)));
end


fclose(file1);
fclose(file2);
fclose(file3);
fclose(file4);
```

File4:-Output of fft:https://github.com/kkousar/HLS/blob/main/output.txt

File3:-ouput of cp_remover: https://github.com/kkousar/HLS/blob/main/output_cp.txt

Vivado HLS:

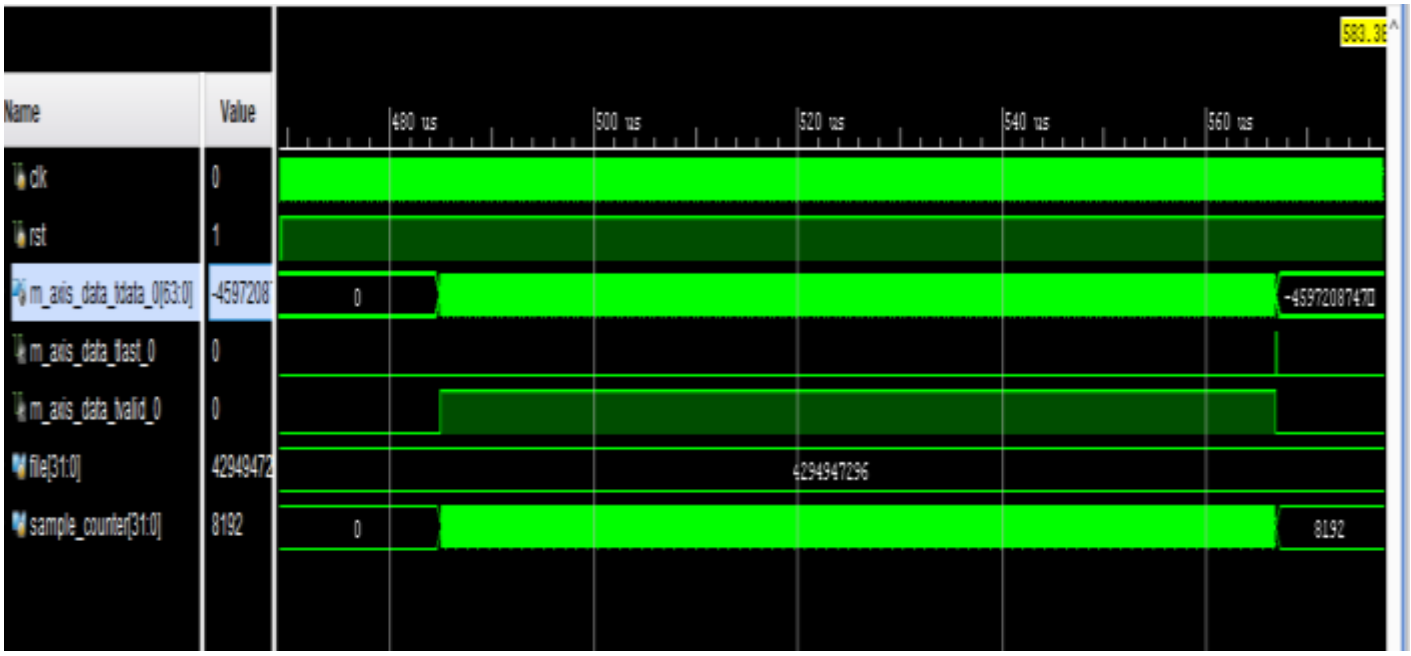


VERILOG TEST BENCH:

```
`timescale 1ns / 1p
module fft_tb(
   );
    reg clk;
       reg rst;
       wire [63:0]m_axis_data_tdata_0;
       wire m_axis_data_tlast_0;
       wire m_axis_data_tvalid_0;
       integer file;
       wire last_0;
       reg [63:0] stored_values [0:8192];
        reg [31:0] sample_counter;

        fft8 fft8_i
            (.ap_clk_0(clk),
             .aresetn_0(rst),
             .m_axis_data_tdata_0(m_axis_data_tdata_0),
             .m_axis_data_tlast_0(m_axis_data_tlast_0),
             .m_axis_data_tvalid_0(m_axis_data_tvalid_0));



           always #5 clk=~clk;
           initial begin
```

```
            rst=0;clk=1;
           // #100 rst=1;
            //#89130000 $finish;
            end

            initial begin

            file=$fopen("out_fft2.txt","w");
            sample_counter = 0;

          while (sample_counter < 8192) begin
           #10;  // Assuming a sampling rate of 10 units

            if (m_axis_data_tvalid_0) begin
             $fwrite(file, "%h\n", m_axis_data_tdata_0);
            stored_values[sample_counter] = m_axis_data_tdata_0;
            sample_counter = sample_counter + 1;
           end
          end
          $fclose(file);
        end
endmodule
```

Output:



Matlab output:

Sample1:1.610222 +i6.042513→40c15c443fce1bc1

Sample2:-3.702850 +i-4.205078→c0869000c06cfb7f

….

vivado fft_ip output:

40c15ba03fce1980

c0869070c06cfd60

Vivado FFT IP output:https://github.com/kkousar/HLS/blob/main/hls%20files/out_fft.txt