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 gen(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
      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){
#pragma HLS INTERFACE axis register both port=out
#pragma HLS INTERFACE axis register both port=in

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

Test bench code:

```
#include <hls_stream.h>
#include <complex>
#include <iostream>
#include <fstream>
using namespace hls;
using namespace std;
typedef complex<float> cmp;
void gen(stream<cmp> &out);
void cp(stream<cmp> &in,stream<cmp> &out);
int main(){
    stream<cmp> out;
    stream<cmp> cp_out;
    gen(out);
    cmp o;
/*  ofstream gen_output("gen_output.dat");
    for (int i=0;i<8800;i++){
        o=out.read();
        gen_output<<o<<endl;
    }*/
    cp(out,cp_out);
    ofstream output("cp_output.dat");
        for (int i=0;i<8192;i++){
            o=cp_out.read();
            output<<o<<endl;
        }
        // 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");
}}
```

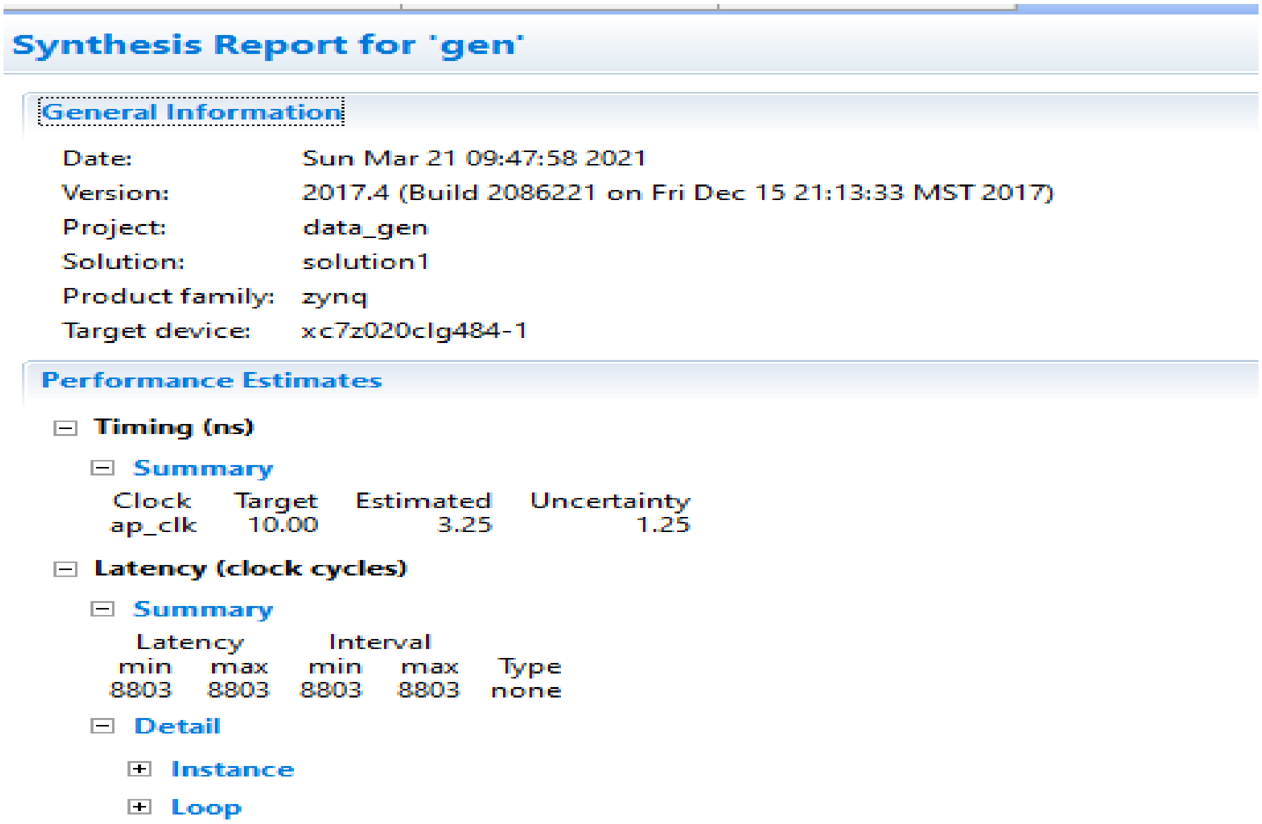
FILES

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

Generator Synthesis report:



Synthesis Report for 'gen'

General Information

Date:	Sun Mar 21 09:47:58 2021
Version:	2017.4 (Build 2086221 on Fri Dec 15 21:13:33 MST 2017)
Project:	data_gen
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)

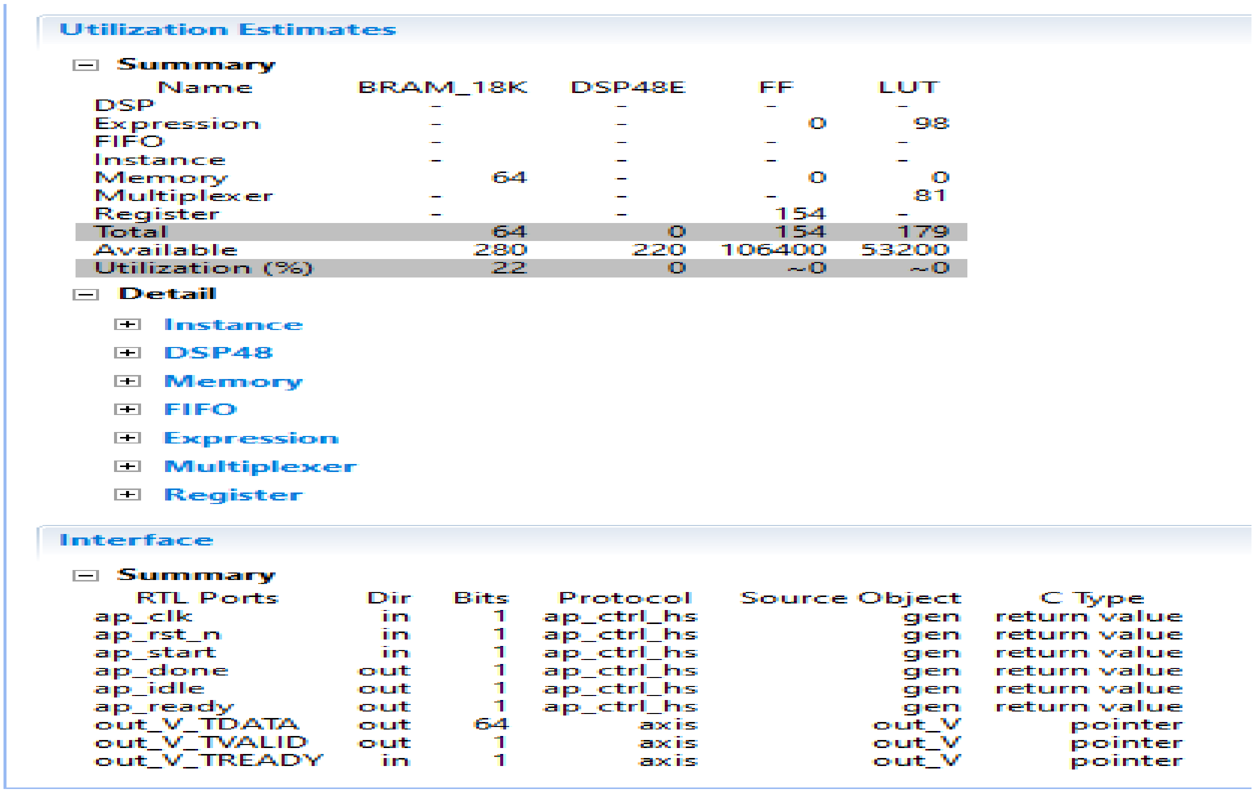
⊟ Summary

Latency		Interval		
min	max	min	max	Type
8803	8803	8803	8803	none

⊟ Detail

⊞ Instance

⊞ Loop



Utilization Estimates

⊟ Summary

Name	BRAM_18K	DSP48E	FF	LUT
DSP	-	-	-	-
Expression	-	-	0	98
FIFO	-	-	-	-
Instance	-	-	-	-
Memory	64	-	0	0
Multiplexer	-	-	-	81
Register	-	-	154	-
Total	64	0	154	179
Available	280	220	106400	53200
Utilization (%)	22	0	~0	~0

⊟ Detail

⊞ Instance

⊞ DSP48

⊞ Memory

⊞ FIFO

⊞ Expression

⊞ Multiplexer

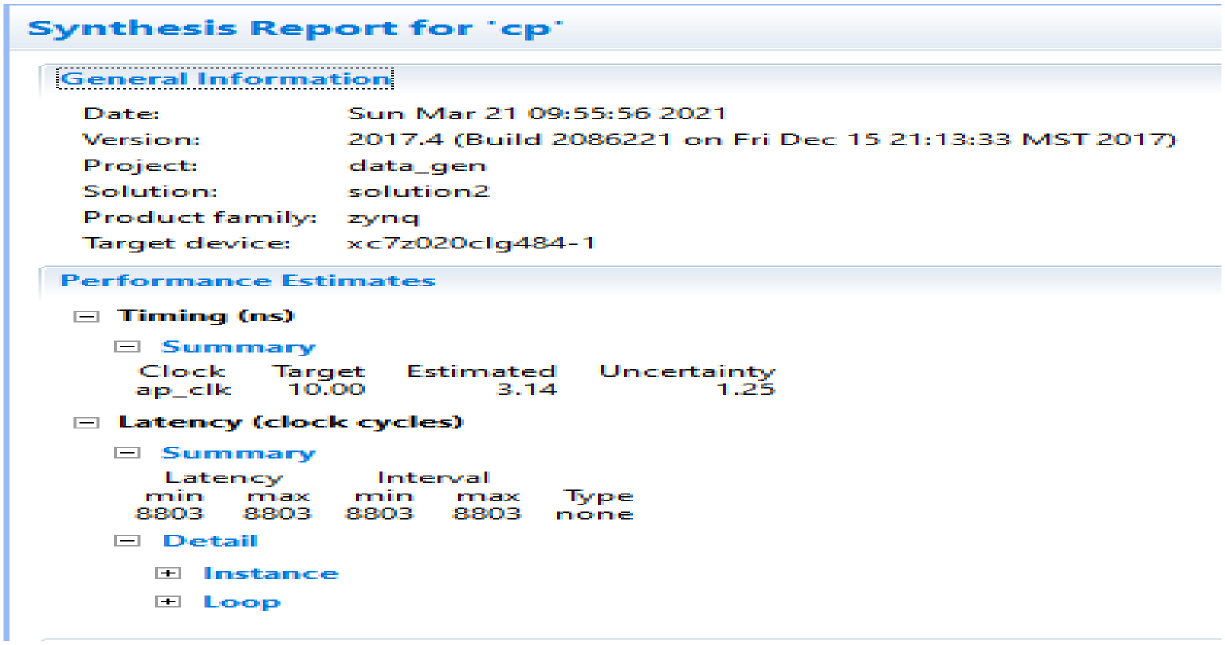
⊞ Register

Interface

⊟ Summary

RTL Ports	Dir	Bits	Protocol	Source Object	C Type
ap_clk	in	1	ap_ctrl_hs	gen	return value
ap_rst_n	in	1	ap_ctrl_hs	gen	return value
ap_start	in	1	ap_ctrl_hs	gen	return value
ap_done	out	1	ap_ctrl_hs	gen	return value
ap_idle	out	1	ap_ctrl_hs	gen	return value
ap_ready	out	1	ap_ctrl_hs	gen	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:	Sun Mar 21 09:55:56 2021
Version:	2017.4 (Build 2086221 on Fri Dec 15 21:13:33 MST 2017)
Project:	data_gen
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)

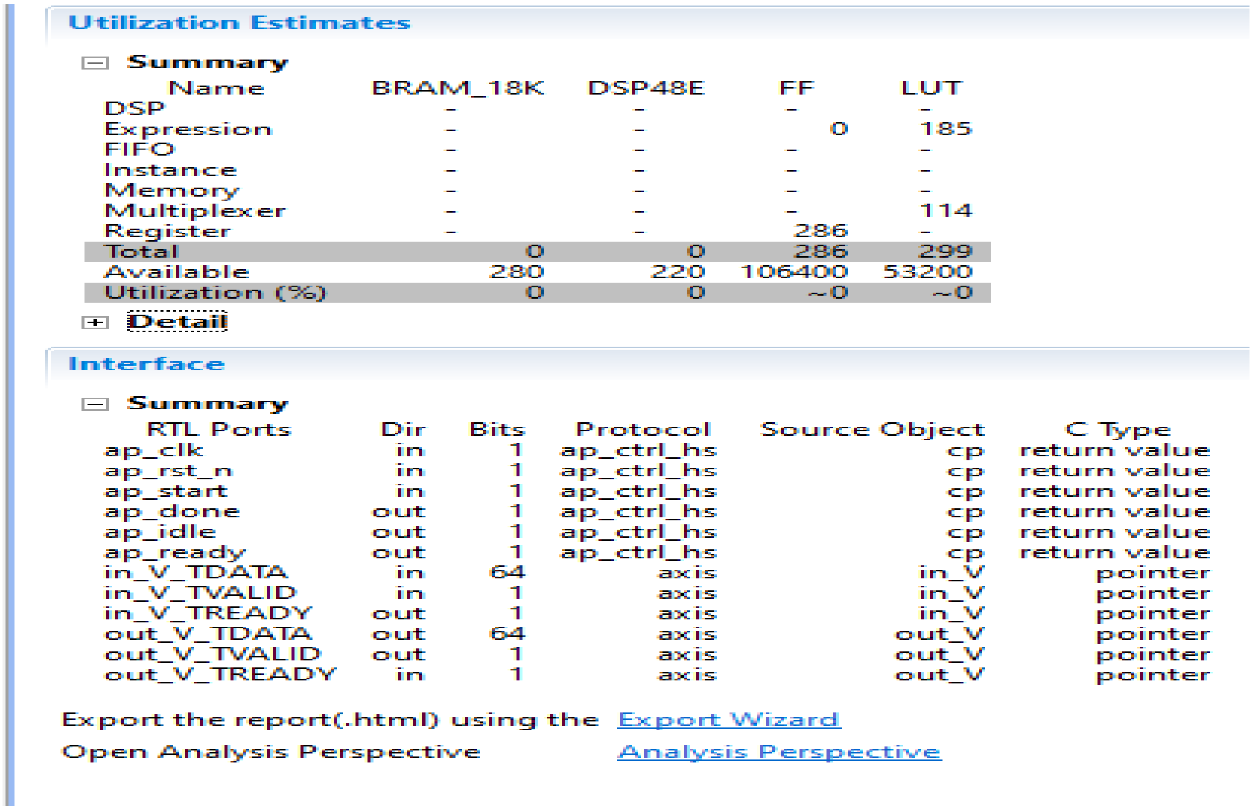
⊟ Summary

Latency		Interval		
min	max	min	max	Type
8803	8803	8803	8803	none

⊟ Detail

⊞ Instance

⊞ Loop



Utilization Estimates

⊟ Summary

Name	BRAM_18K	DSP48E	FF	LUT
DSP	-	-	-	-
Expression	-	-	0	185
FIFO	-	-	-	-
Instance	-	-	-	-
Memory	-	-	-	-
Multiplexer	-	-	-	114
Register	-	-	286	-
Total	0	0	286	299
Available	280	220	106400	53200
Utilization (%)	0	0	~0	~0

⊞ Detail

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

Export the report(.html) using the Export Wizard

Open Analysis Perspective Analysis Perspective

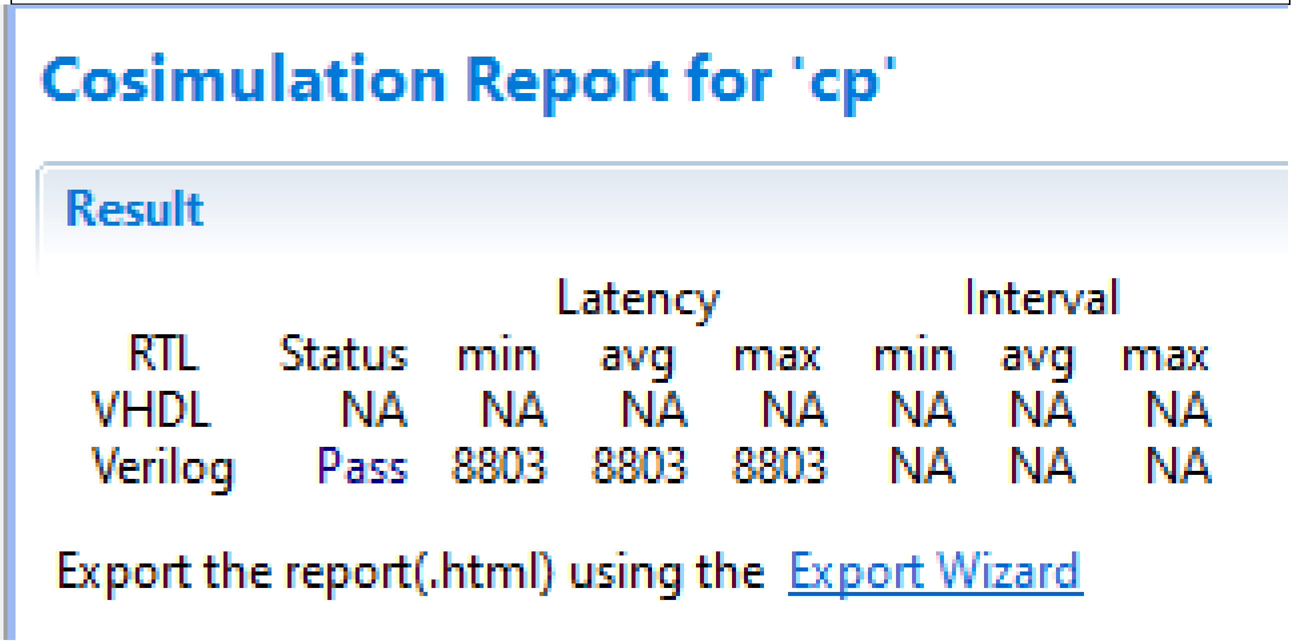
Simulation report

```
INFO: [SIM 2] *************** CSIM start ***************
INFO: [SIM 4] CSIM will launch GCC as the compiler.
   Compiling ../../../../../../../Downloads/HLS_Vivado/tb.cpp in debug
mode
   Compiling ../../../generator.cpp in debug mode
   Compiling ../../../cp_remover.cpp in debug mode
   Generating csim.exe
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 Sun May 21 23:44:12 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	8803	8803	8803	NA	NA	NA

Export the report(.html) using the Export Wizard

Matlab code:

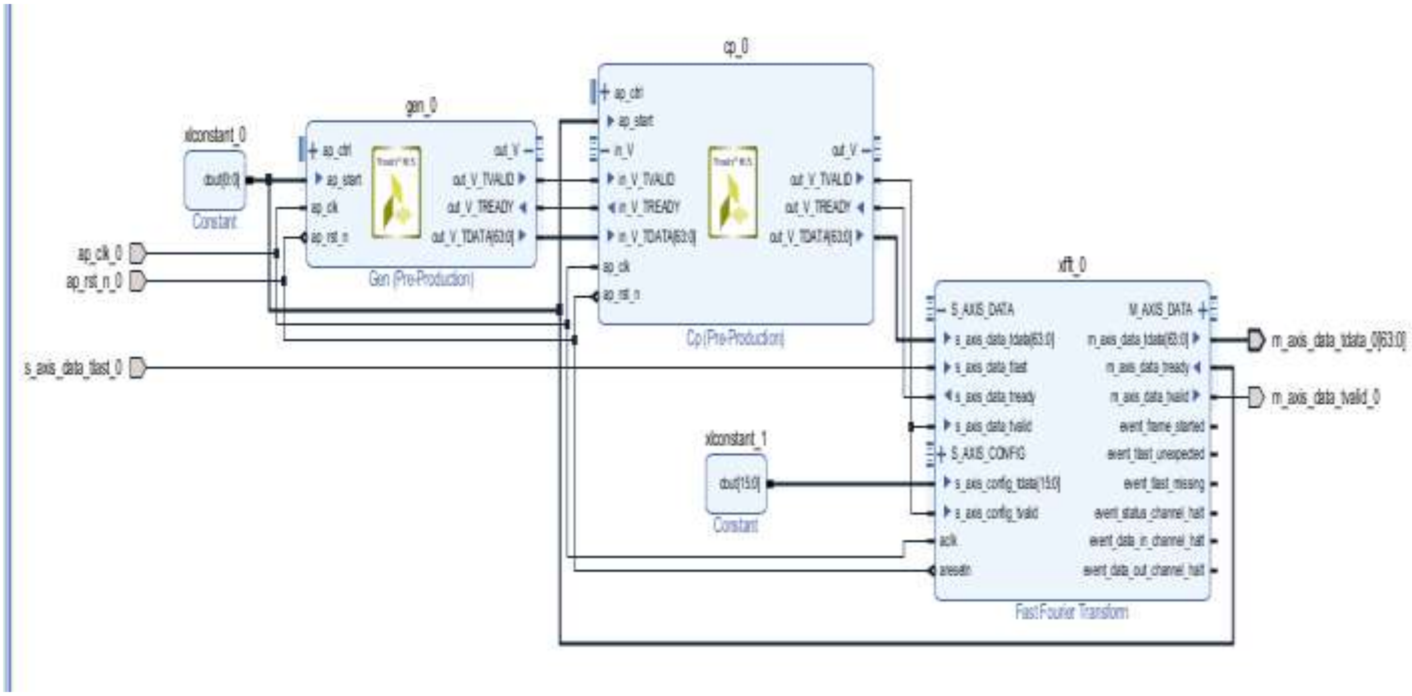
```
file1 = fopen('/MATLAB Drive/puschTxAfterChannelImag.txt','r')
file2 = fopen('/MATLAB Drive/puschTxAfterChannelReal.txt','r')
file3=fopen('output.txt','w')
file4=fopen('output_cp.txt','w')

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

cp_l = [320, 288];
for i = 1:numel(cp_l)
    for j = 1:4096
        cp_removed = [cp_removed; cp_added(cp_l(i) + j + c)];
    end
    c = c + 288;
```

```
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 cp-remover:https://github.com/kkousar/HLS/blob/main/output.txt

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

Vivado HLS:



VERILOG TEST BENCH:

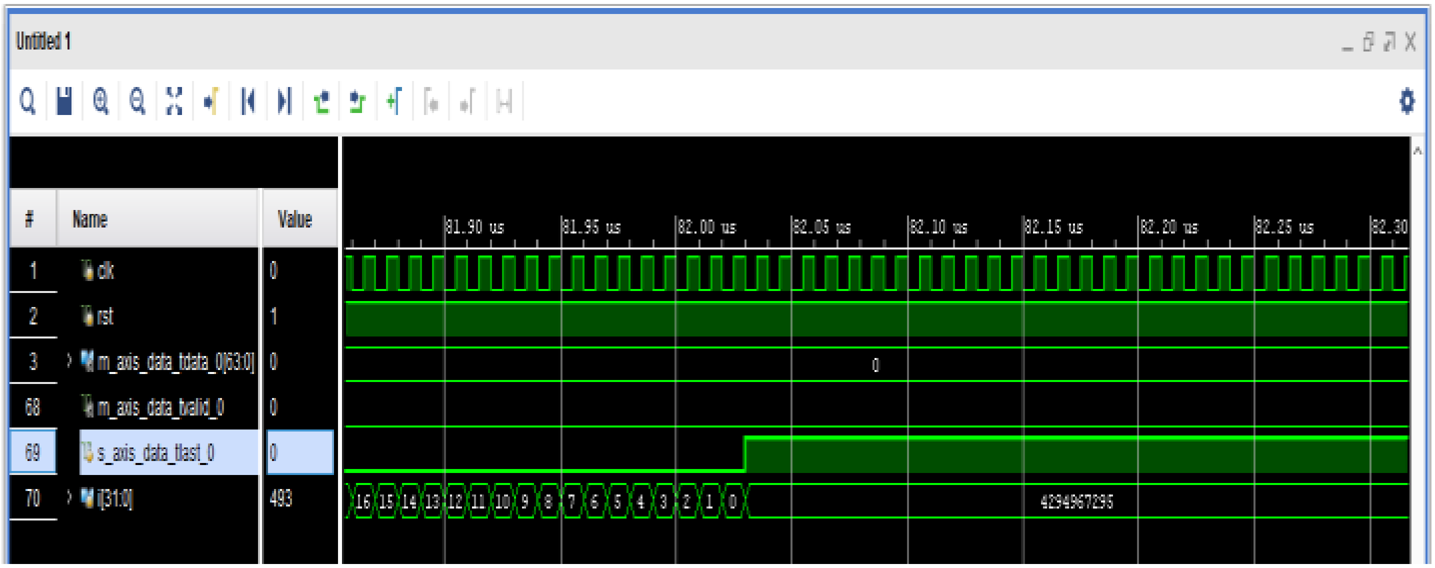
```
`timescale 1ns / 1ps

//////////////////////////////////////////////////////////////////////////////////
```

```
// Company:

// Engineer:

//

// Create Date: 05/21/2023 04:15:29 PM

// Design Name:

// Module Name: cp_new_tb

// Project Name:

// Target Devices:

// Tool Versions:

// Description:

//

// Dependencies:

//

// Revision:

// Revision 0.01 - File Created

// Additional Comments:

//

//////////////////////////////////////////////////////////////////////////////////

module cp_new_tb();

reg clk;

reg rst;

wire [63:0]m_axis_data_tdata_0;//output data of fft

wire m_axis_data_tvalid_0;//valid output of fft

reg s_axis_data_tlast_0;//tlast input to fft
```

```
fft_new fft_new_i
      (.ap_clk_0(clk),
       .ap_rst_n_0(rst),
       .m_axis_data_tdata_0(m_axis_data_tdata_0),
       .m_axis_data_tvalid_0(m_axis_data_tvalid_0));
always #5 clk=~clk;//clock with 10 unit period
initial begin
clk=0;
rst=0;
#10 rst=1;
s_axis_data_tlast_0=1'b0;
end
integer i=0;
initial
begin
   #100
   for (i=8192;i>=0;i=i-1)begin
   #10
      if (i==0)begin
         s_axis_data_tlast_0=1'b1;//set last signal high when the last data form the stream is accessed
      end
   end
 end
initial begin
   #100000; // Additional delay after setting s_axis_data_tvalid_1 to 0
```

```
    s_axis_data_tlast_0=1'b0;

   $finish; // End the simulation

  end

endmodule
```

Output:



Untitled 1

#	Name	Value
1	clk	0
2	rst	1
3	m_axis_data_tdata_0[63:0]	0
68	m_axis_data_tvalid_0	0
69	s_axis_data_tlast_0	0
70	i[31:0]	493