V.GOKUL KUMAR
FUTURE WIRELESS COMMUNICATIONS (FWC)
Indian Institute of Technology Hyderabad
Email: velicharla@outlook.com

Date:26-05-2023

Hls ASSIGNMENT-8

Cyclic Prefix removal in 5G NR

Vivado/HLS Assignment

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.

Vivado/HLS Guidelines:

1. Use AXI streaming interface for the input and output ports of all your modules.
2. Use three data files in HLS testbenches that you use: input, output and reference output.
    a. Input for reading inputs.
    b. Output for storing the generated outputs.
    c. Reference outputs for finding the error and for deciding if it's tolerable based on the precision required. Take 10e-3 as precision required.
3. Go through UG902 Vivado HLS user guide if you have any doubts regarding HLS.
4. Go through UG893 Vivado user guide if you have any doubts regarding Vivado.
5. Go through PG109 Xilinx FFT IP guide if you have any doubts regarding FFT IP.
6. Use Google if you have any other doubts.

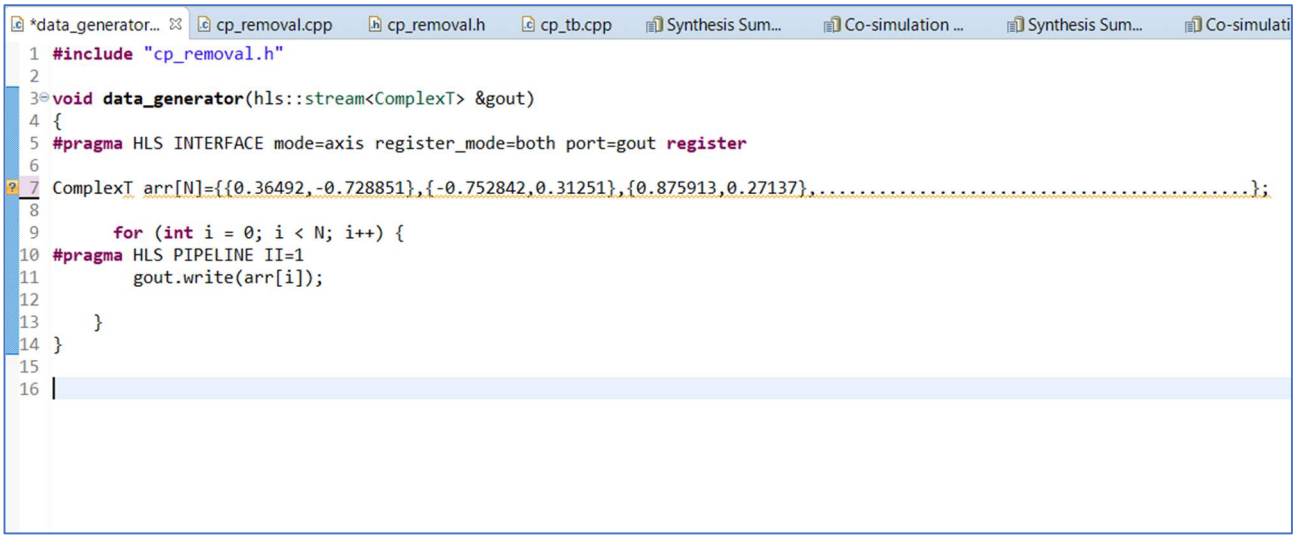
Configuration for the System:

1. *Total number of symbols: 2;*
2. *First symbol CP length: 320;*
3. *Second symbol CP length: 288;*
4. *CP Input length for first symbol: 4096+320;*
5. *CP Input length for second symbol: 4096+288;*
6. *CP Output length for both the symbols: 4096;*
7. *FFT Input length: 4096;*
8. *FFT Output length: 4096;*

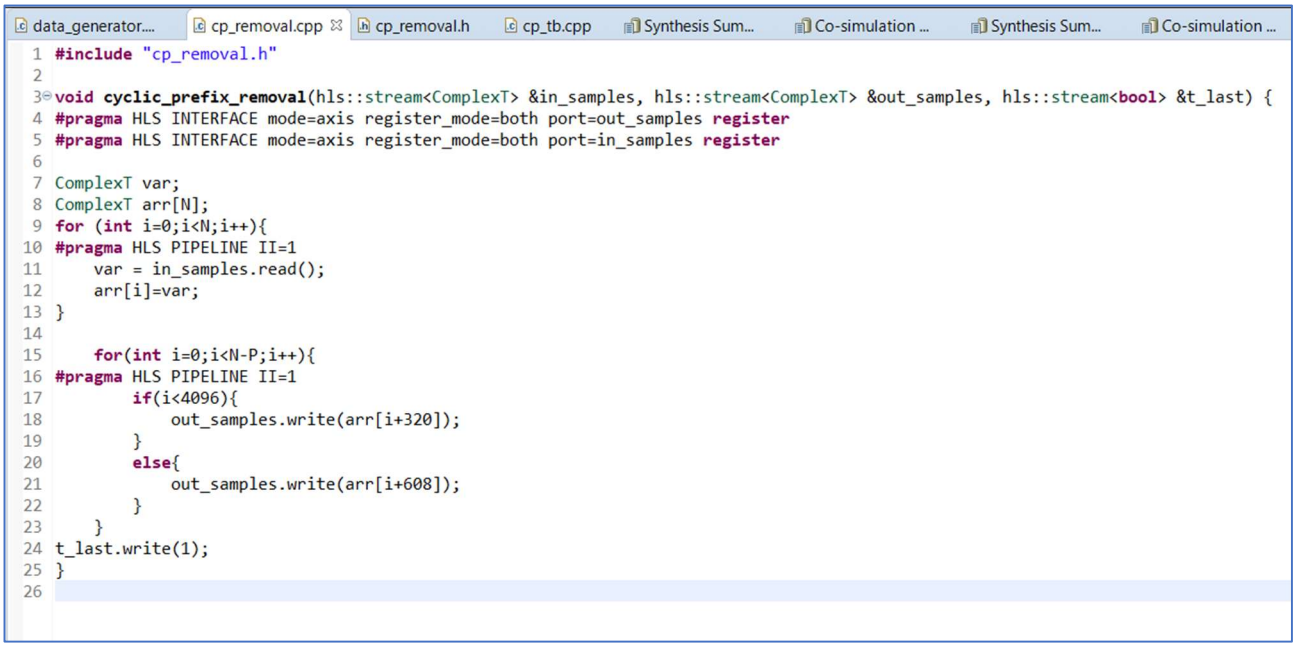
HLS:

Design:

Main function: (Data Generator)



Main function: (Cyclic Prefix Remover)



Header File:

data_generator.... | cp_removal.cpp | cp_removal.h | cp_tb.cpp | Synthesis Sum... | Co-simulation ... | Synthesis Sum... | Co-simulation ...

```
#include <iostream>
#include <complex>
#include <fstream>
#include <hls_stream.h>
#include <vector>
using namespace std;
#define N 8800
#define P 608

#define CP_LEN_1 320
#define CP_LEN_2 288
#define CP_INPUT_LEN_1 (4096 + CP_LEN_1)
#define CP_INPUT_LEN_2 (4096 + CP_LEN_2)
#define CP_OUTPUT_LEN 4096


typedef struct {
    float real;
    float imag;
} ComplexT;

void data_generator(hls::stream<ComplexT> & gout);
void cyclic_prefix_removal(hls::stream<ComplexT> &in_samples, hls::stream<ComplexT> &out_samples, hls::stream<bool> &t_last);

```

Self-checking Test Bench:

data_generator.... | cp_removal.cpp | cp_removal.h | cp_tb.cpp | Synthesis Sum... | Co-simulation ...

```
#include "cp_removal.h"

int main() {

    ComplexT arr[N];
    ComplexT arr2[N];

    float a,b;
    int retval=0;

    ifstream File1("puschTxAfterChannelReal.txt");
    ifstream File2("puschTxAfterChannelImag.txt");
    for (int i = 0; i < N; i++) {
        File1 >> a;
        File2 >> b;
        arr[i].real = a;
        arr[i].imag=b;
    }
    File1.close();
    File2.close();

    for (int i = 0; i < CP_OUTPUT_LEN; i++) {
      arr2[i] = arr[i + CP_LEN_1];
    }

    for (int i = 0; i < CP_OUTPUT_LEN; i++) {
        arr2[i + CP_OUTPUT_LEN] = arr[i + CP_INPUT_LEN_1 + CP_LEN_2];
    }

    ofstream File3("Reference.dat");
    for (int i = 0; i < N-P; i++){
    File3<<"{"<<arr2[i].real<<","<<arr2[i].imag<<"}"<<endl;
    }
    File3.close();



    ofstream File4("output.dat");
```

```
39
40      ComplexT arr3[N];
41      hls::stream<ComplexT> gdata,cpdata;
42      hls::stream<bool> last;
43
44      ComplexT output;
45      hls::stream<int> z;
46      int t=0;
47
48      data_generator(gdata);
49      cyclic_prefix_removal(gdata, cpdata, last);
50
51          for (int i=0;i<N-P;i++){
52          if (!cpdata.empty()) {
53            output = cpdata.read();
54            arr3[i].real=output.real;
55            arr3[i].imag=output.imag;
56            File4<<"{"<<arr3[i].real<<","<<arr3[i].imag<<"}"<<endl;
57          }
58
59      }
60      File4.close();
61
62      // can be modified further for testing 10e-3 but unnecessary
63      // Compare the results file with the golden results
64      retval = system("diff --brief -w reference.dat output.dat");
65      if (retval != 0) {
66          printf("Failed\n");
67          retval=1;
68      } else {
69          printf("Matched with reference output !\n");
70    }
71       return 0;
72  }
```

Input Files:

Imaginary Samples:

puschTxAfterChannelImag.txt | data_generator.cpp

```
1  -0.728851
2  0.312510
3  0.271370
4  -0.787211
5  0.793227
6  -0.345721
7  -0.358169
8  0.705594
9  -0.701969
10 0.334532
11 0.206578
12 -0.688014
13 0.713312
14 -0.454452
15 -0.075746
16 0.583362
17 -0.710587
18 0.369182
19 -0.002131
20 -0.466399
21 0.621525
22 -0.368473
23 -0.118023
24 0.551553
25 -0.523579
26 0.298048
27 0.113472
28 -0.465206
29 0.382129
30 -0.242162
31 -0.126200
32 0.347238
33 -0.289305
34 0.138652
35 0.241981
36 -0.377481
37 0.268369
38 0.050652
```

Real Samples:

puschTxAfterChannelReal.txt | puschTxAfterChannelI

```
1  0.364920
2  -0.752842
3  0.875913
4  -0.369930
5  -0.284297
6  0.775180
7  -0.743295
8  0.341801
9  0.200641
10 -0.657016
11 0.732162
12 -0.371676
13 -0.233051
14 0.573324
15 -0.643119
16 0.433066
17 0.082850
18 -0.603719
19 0.742388
20 -0.394873
21 -0.111544
22 0.526113
23 -0.602560
24 0.324726
25 0.094749
26 -0.376094
27 0.488746
28 -0.286551
29 -0.074839
30 0.355682
31 -0.404309
32 0.196520
33 0.138709
34 -0.369935
35 0.300978
36 -0.061944
37 -0.196167
38 0.383260
```

Reference File: (golden data)

data_generat... | cp_removal.cpp | Reference.dat | cp_removal.h | cp_tb.cpp

```
{-0.601084,-0.059909}
{0.571592,0.530777}
{0.096652,-0.704768}
{-0.493203,0.428746}
{0.595521,0.033847}
{-0.425508,-0.530197}
{-0.015958,0.623015}
{0.504235,-0.433297}
{-0.612428,-0.060689}
{0.46423,0.484835}
{0.038207,-0.673277}
{-0.418757,0.335124}
{0.506749,0.073827}
{-0.383995,-0.44953}
{-0.017262,0.549452}
{0.395049,-0.282604}
{-0.398053,-0.061566}
{0.20797,0.298626}
{0.060601,-0.301795}
{-0.372701,0.091815}
{0.187062,0.178716}
{-0.017487,-0.13696}
{-0.173727,0.156696}
{0.204019,0.009835}
{-0.09403,-0.262932}
{-0.244866,0.240371}
{0.351761,0.030759}
{-0.129693,-0.372167}
{-0.081731,0.435934}
{0.371914,-0.249935}
{-0.612817,-0.196231}
{0.186924,0.512677}
{0.181238,-0.62866}
{-0.562053,0.220645}
{0.738443,0.218542}
{-0.426321,-0.66548}
{-0.163477,0.665213}
{0.667362,-0.427728}
```

Output File: (data returned by function)

output.dat | Reference.dat | cp_tb.cpp | Synthesis S... | Co-simulatio...

```
{-0.601084,-0.059909}
{0.571592,0.530777}
{0.096652,-0.704768}
{-0.493203,0.428746}
{0.595521,0.033847}
{-0.425508,-0.530197}
{-0.015958,0.623015}
{0.504235,-0.433297}
{-0.612428,-0.060689}
{0.46423,0.484835}
{0.038207,-0.673277}
{-0.418757,0.335124}
{0.506749,0.073827}
{-0.383995,-0.44953}
{-0.017262,0.549452}
{0.395049,-0.282604}
{-0.398053,-0.061566}
{0.20797,0.298626}
{0.060601,-0.301795}
{-0.372701,0.091815}
{0.187062,0.178716}
{-0.017487,-0.13696}
{-0.173727,0.156696}
{0.204019,0.009835}
{-0.09403,-0.262932}
{-0.244866,0.240371}
{0.351761,0.030759}
{-0.129693,-0.372167}
{-0.081731,0.435934}
{0.371914,-0.249935}
{-0.612817,-0.196231}
{0.186924,0.512677}
{0.181238,-0.62866}
{-0.562053,0.220645}
{0.738443,0.218542}
{-0.426321,-0.66548}
{-0.163477,0.665213}
{0.667362,-0.427728}
```

C simulation printed output:

data_generator.cpp | cp_removal.cpp | cp_removal.h | cp_tb.cpp | Synthesis Summary... | Co-simulation Repo... | Synthesis Summary...

```
1 INFO: [SIM 2] *************** CSIM start ***************
2 INFO: [SIM 4] CSIM will launch GCC as the compiler.
3    Compiling ../../../../vitis-hls/Assignment_8/HLS/Source/cp_removal.cpp in debug mode
4    Generating csim.exe
5 Matched with reference output !
6 WARNING [HLS SIM]: hls::stream 'hls::stream<bool, 0>0' contains leftover data, which may result in RTL simulation hanging.
7 INFO [HLS SIM]: The maximum depth reached by any hls::stream() instance in the design is 8800
8 INFO: [SIM 1] CSim done with 0 errors.
9 INFO: [SIM 3] *************** CSIM finish ***************
10
```

Synthesis Report:

Data Generator:

data_generator.cpp | cp_removal.cpp | cp_removal.h | cp_tb.cpp | Synthesis Summary(solutio... | Co-simulation Report(soluti... | Synthesis Summary(solutio... | Co-simulation Report(soluti...

Synthesis Summary Report of 'data_generator'

General Information

Date:	Fri May 26 18:06:35 2023	Solution:	solution1 (Vivado IP Flow Target)
Version:	2022.2 (Build 3670227 on Oct 13 2022)	Product family:	zynq
Project:	project_38	Target device:	xc7z020-clg484-1

Timing Estimate

Target	Estimated	Uncertainty
10.00 ns	3.797 ns	2.70 ns

Performance & Resource Estimates

% Modules Loops

Modules & Loops	Issue Type	Violation Type	Distance	Slack	Latency(cycles)	Latency(ns)	Iteration Latency	Interval	Trip Count	Pipelined	BRAM	DSP	FF	LUT	URAM
data_generator				-	8803	8.803E4	-	8804	-	no	64	0	19	84	0
VITIS_LOOP_1768_1				-	8801	8.801E4	3	1	8800	yes	-	-	-	-	-

Performance Pragma

Modules & Loops	Target TI(cycles)	TI(cycles)	TI met
data_generator	-	-	-
VITIS_LOOP_1768_1	-	-	-

HW Interfaces

AXIS

Interface	Register Mode	TDATA	TREADY	TVALID
gout	both	64	1	1

TOP LEVEL CONTROL

Interface	Type	Ports
ap_clk	clock	ap_clk
ap_rst_n	reset	ap_rst_n
ap_ctrl	ap_ctrl_hs	ap_done ap_idle ap_ready ap_start

SW I/O Information

Top Function Arguments

Argument	Direction	Datatype
gout	out	stream<ComplexT 0>&

SW-to-HW Mapping

Argument	HW Interface	HW Type
gout	gout	interface

Pragma Report

Argument	HW Interface	HW Type
gout	gout	interface

Pragma Report

Valid Pragma Syntax

Type	Options	Location	Function
interface	mode=axis register_mode=both port=out_samples	vitis-hls/Assignment_8/HLS/Source/cp_removal.cpp:4	cyclic_prefix_removal
interface	mode=axis register_mode=both port=in_samples	vitis-hls/Assignment_8/HLS/Source/cp_removal.cpp:5	cyclic_prefix_removal
pipeline	II=1	vitis-hls/Assignment_8/HLS/Source/cp_removal.cpp:10	cyclic_prefix_removal
pipeline	II=1	vitis-hls/Assignment_8/HLS/Source/cp_removal.cpp:16	cyclic_prefix_removal
interface	mode=axis register_mode=both port=gout register	vitis-hls/Assignment_8/HLS/Source/data_generator.cpp:5	data_generator
pipeline	II=1	vitis-hls/Assignment_8/HLS/Source/data_generator.cpp:1769	data_generator

Bind Op Report

No filter settings

Name	DSP	Pragma	Variable	Op	Impl	Latency
data_generator	-					
VITIS_LOOP_1768_1						

No user config_op information

Bind Storage Report

No filter settings

Cyclic Prefix Remover：

data_generator.cpp | cp_removal.cpp | cp_removal.h | cp_tb.cpp | Synthesis Summary... | Co-simulation Repo... | Synthesis Summary... | Co-simulation Repo... | data_generator_csim...

Synthesis Summary Report of 'cyclic_prefix_removal'

General Information

Date: Fri May 26 18:16:40 2023
Version: 2022.2 (Build 3670227 on Oct 13 2022)
Project: project_38
Solution: solution2 (Vivado IP Flow Target)
Product family: zynq
Target device: xc7z020-clg484-1

Timing Estimate

Target	Estimated	Uncertainty
10.00 ns	5.463 ns	2.70 ns

Performance & Resource Estimates

Modules Loops

Modules & Loops	Issue Type	Violation Type	Distance	Slack	Latency(cycles)	Latency(ns)	Iteration Latency	Interval	Trip Count	Pipelined	BRAM	DSP	FF	LUT	URAM
cyclic_prefix_removal				-	17002	1.700E5	-	17003	-	no	64	0	43	374	0
cyclic_prefix_removal_Pipeline_VITIS_LOOP_9_1				-	8802	8.802E4	-	8802	-	no	0	0	16	69	0
cyclic_prefix_removal_Pipeline_VITIS_LOOP_15_2				-	8194	8.194E4	-	8194	-	no	0	0	18	167	0

Performance Pragma

Modules & Loops	Target TI(cycles)	TI(cycles)	TI met
cyclic_prefix_removal	-	-	-
cyclic_prefix_removal_Pipeline_VITIS_LOOP_9_1	-	-	-
cyclic_prefix_removal_Pipeline_VITIS_LOOP_15_2	-	-	-

HW Interfaces

AXIS

Interface	Register Mode	TDATA	TREADY	TVALID
in_samples	both	64	1	1
out_samples	both	64	1	1

AP_FIFO

Interface	Data Width
t_last	1

TOP LEVEL CONTROL

Interface	Type	Ports
ap_clk	clock	ap_clk
ap_rst_n	reset	ap_rst_n
ap_ctrl	ap_ctrl_hs	ap_done ap_idle ap_ready ap_start

SW I/O Information

Top Function Arguments

Argument	Direction	Datatype
in_samples	in	stream<ComplexT 0>&
out_samples	out	stream<ComplexT 0>&
t_last	out	stream<bool 0>&

SW-to-HW Mapping

Argument	HW Interface	HW Type
in_samples	in_samples	interface
out_samples	out_samples	interface
t_last	t_last	interface

Pragma Report

Valid Pragma Syntax

Type	Options	Location	Function
interface	mode=axis register_mode=both port=out_samples	vitis-hls/Assignment_8/HLS/Source/cp_removal.cpp:4	cyclic_prefix_removal
interface	mode=axis register_mode=both port=in_samples	vitis-hls/Assignment_8/HLS/Source/cp_removal.cpp:5	cyclic_prefix_removal
pipeline	II=1	vitis-hls/Assignment_8/HLS/Source/cp_removal.cpp:10	cyclic_prefix_removal
pipeline	II=1	vitis-hls/Assignment_8/HLS/Source/cp_removal.cpp:16	cyclic_prefix_removal
interface	mode=axis register_mode=both port=gout register	vitis-hls/Assignment_8/HLS/Source/data_generator.cpp:5	data_generator
pipeline	II=1	vitis-hls/Assignment_8/HLS/Source/data_generator.cpp:1769	data_generator

Bind Op Report

No filter settings

Name	DSP	Pragma	Variable	Op	Impl	Latency
cyclic_prefix_removal	-					
cyclic_prefix_removal_Pipeline_VITIS_LOOP_9_1	-					
cyclic_prefix_removal_Pipeline_VITIS_LOOP_15_2	-					

Co-simulation printed output:

Data Generator:

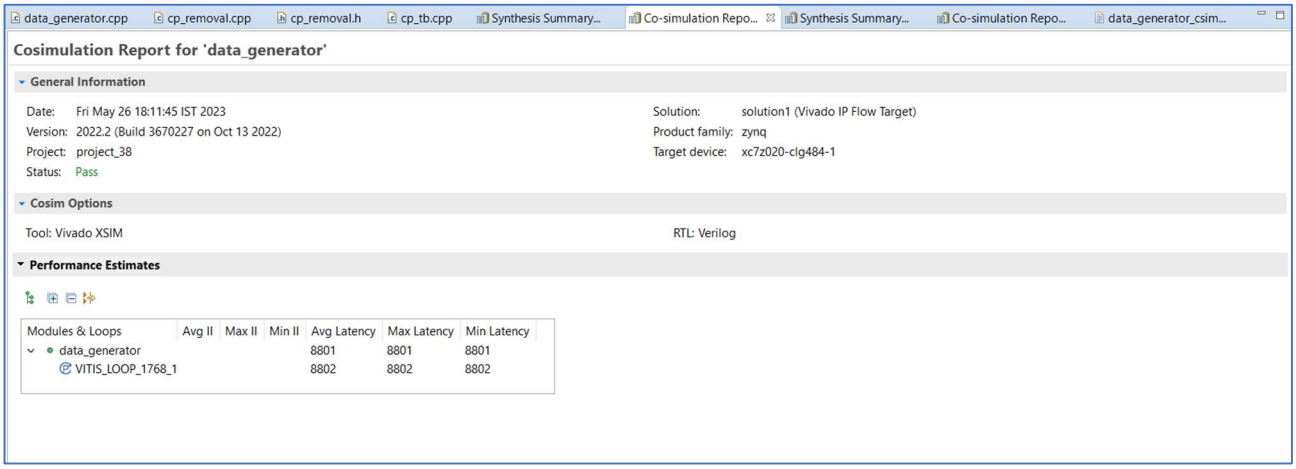
```
////////////////////////////////////////////////////////////////////////////////////
// Inter-Transaction Progress: Completed Transaction / Total Transaction
// Intra-Transaction Progress: Measured Latency / Latency Estimation * 100%
//
// RTL Simulation : "Inter-Transaction Progress" ["Intra-Transaction Progress"] @ "Simulation Time"
////////////////////////////////////////////////////////////////////////////////////
// RTL Simulation : 0 / 1 [0.00%] @ "125000"
// RTL Simulation : 1 / 1 [100.00%] @ "88155000"
////////////////////////////////////////////////////////////////////////////////////
$finish called at time : 88215 ns : File "C:/Users/velic/OneDrive/Desktop/project_38/solution1/sim/verilog/data_generator.autotb.v" Line 210
## quit
INFO: [Common 17-206] Exiting xsim at Fri May 26 18:11:38 2023...
INFO: [COSIM 212-316] Starting C post checking ...
Matched with reference output !
WARNING [HLS SIM]: hls::stream 'hls::stream<bool, 0>0' contains leftover data, which may result in RTL simulation hanging.
INFO [HLS SIM]: The maximum depth reached by any hls::stream() instance in the design is 8800
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 t
INFO: [HLS 200-111] Finished Command cosim_design CPU user time: 3 seconds. CPU system time: 2 seconds. Elapsed time: 124.974 seconds; current allocated memory: 8
INFO: [HLS 200-112] Total CPU user time: 6 seconds. Total CPU system time: 4 seconds. Total elapsed time: 143.449 seconds; peak allocated memory: 107.293 MB.
Finished C/RTL cosimulation.
```

Cyclic Prefix Remover:

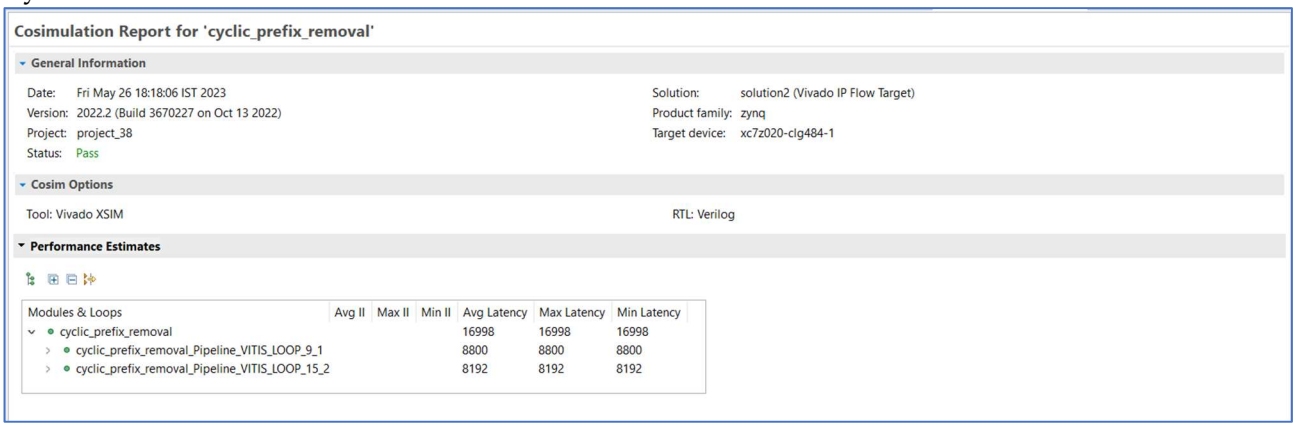
```
////////////////////////////////////////////////////////////////////////////////////
// Inter-Transaction Progress: Completed Transaction / Total Transaction
// Intra-Transaction Progress: Measured Latency / Latency Estimation * 100%
//
// RTL Simulation : "Inter-Transaction Progress" ["Intra-Transaction Progress"] @ "Simulation Time"
////////////////////////////////////////////////////////////////////////////////////
// RTL Simulation : 0 / 1 [0.00%] @ "125000"
// RTL Simulation : 1 / 1 [100.00%] @ "170125000"
////////////////////////////////////////////////////////////////////////////////////
$finish called at time : 170185 ns : File "C:/Users/velic/OneDrive/Desktop/project_38/solution2/sim/verilog/cyclic_prefix_removal.autotb.v" Line 301
## quit
INFO: [Common 17-206] Exiting xsim at Fri May 26 18:18:04 2023...
INFO: [COSIM 212-316] Starting C post checking ...
Matched with reference output !
WARNING [HLS SIM]: hls::stream 'hls::stream<bool, 0>0' contains leftover data, which may result in RTL simulation hanging.
INFO [HLS SIM]: The maximum depth reached by any hls::stream() instance in the design is 8800
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
INFO: [HLS 200-111] Finished Command cosim_design CPU user time: 2 seconds. CPU system time: 1 seconds. Elapsed time: 53.327 seconds; current allocated memory: 8.852 MB.
INFO: [HLS 200-112] Total CPU user time: 4 seconds. Total CPU system time: 2 seconds. Total elapsed time: 66.113 seconds; peak allocated memory: 105.633 MB.
Finished C/RTL cosimulation.
```

Co-simulation Report:

Data Generator:

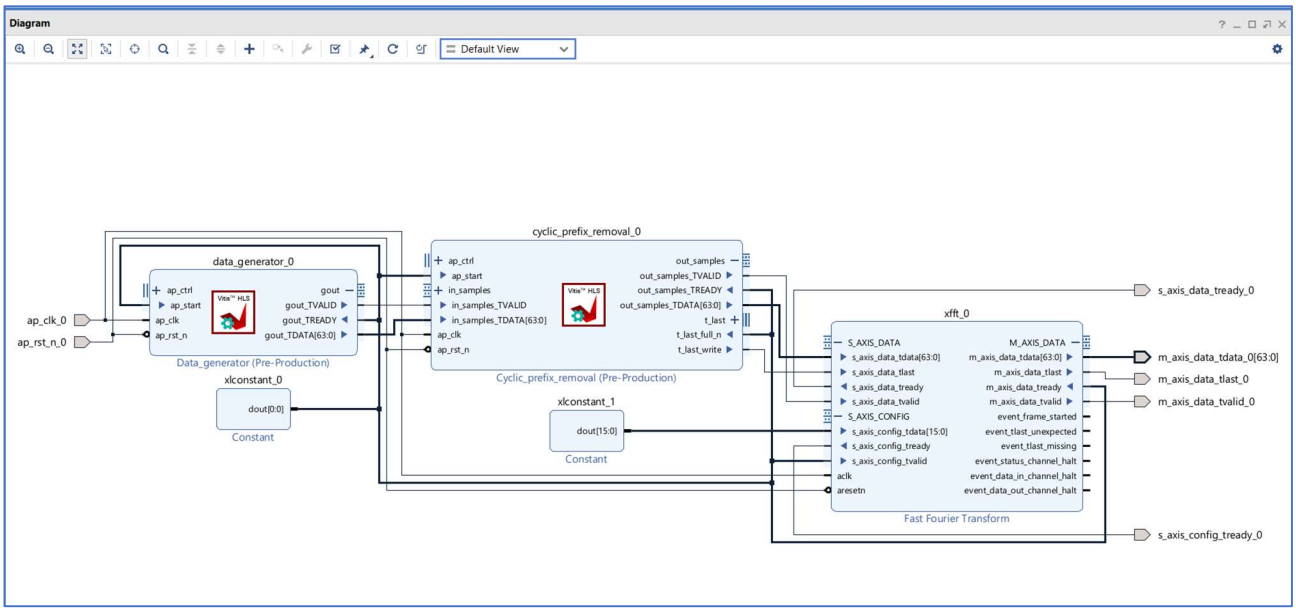


Cyclic Prefix Remover:



VIVADO:

Block Design:



Test Bench:

cp_fft_tb.v | design_1_wrapper.v | cyclic_prefix_removal_hls_deadlock_kernel_monitor_top.vh

C:/Users/velic/project_14/project_14.srcs/sim_1/new/cp_fft_tb.v

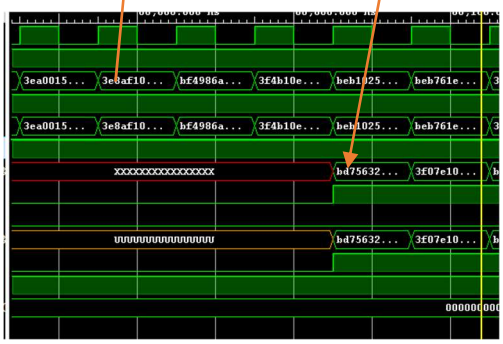
```
1  `timescale 1ns / 1ps
2
3  module cp_fft_tb(
4      );
5
6      `timescale 1ns / 1ps
7
8    reg ap_clk_0;
9    reg ap_rst_n_0;
10   wire [63:0]m_axis_data_tdata_0;
11   wire m_axis_data_tlast_0;
12   wire m_axis_data_tvalid_0;
13   wire s_axis_config_tready_0;
14   wire s_axis_data_tready_0;
15   integer file_handle;
16   reg [63:0] outdata;
17   reg [32:0] counter=0;
18
19 design_1_wrapper in1
20     (.ap_clk_0(ap_clk_0),
21      .ap_rst_n_0(ap_rst_n_0),
22      .m_axis_data_tdata_0(m_axis_data_tdata_0),
23      .m_axis_data_tlast_0(m_axis_data_tlast_0),
24      .m_axis_data_tvalid_0(m_axis_data_tvalid_0),
25      .s_axis_config_tready_0(s_axis_config_tready_0),
26      .s_axis_data_tready_0(s_axis_data_tready_0));
27
28
29          always#5 ap_clk_0=~ap_clk_0;
30
```

```
31
32          initial begin
33          file_handle = $fopen("fft_ip.txt","w");
34          ap_clk_0=0;
35          ap_rst_n_0=0;
36          #10 ap_rst_n_0=1;
37          end
38
39            // Store samples of outdata into a file whenever clock and valid are high
40    always @(posedge ap_clk_0) begin
41      if (m_axis_data_tvalid_0 && ap_clk_0) begin
42      outdata=m_axis_data_tdata_0;
43          $fwrite(file_handle, "%h\n", outdata);
44           counter=counter+1;
45            $display( "%h %d\n", outdata,counter);
46           if(counter==8192)begin
47       $fclose(file_handle);
48  end
49   end
50    end
51
52  endmodule
```

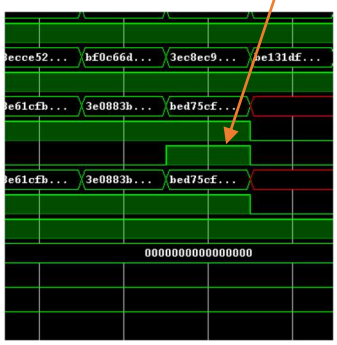
Simulation Window:

Data generator output

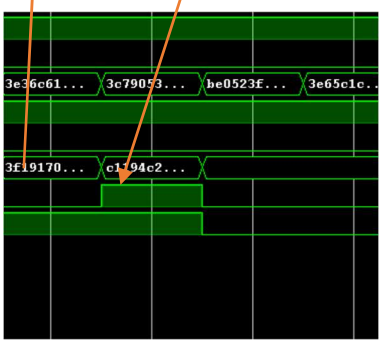
FFT Samples



CP REMOVER block output
out_samples_TVALID (HIGH)

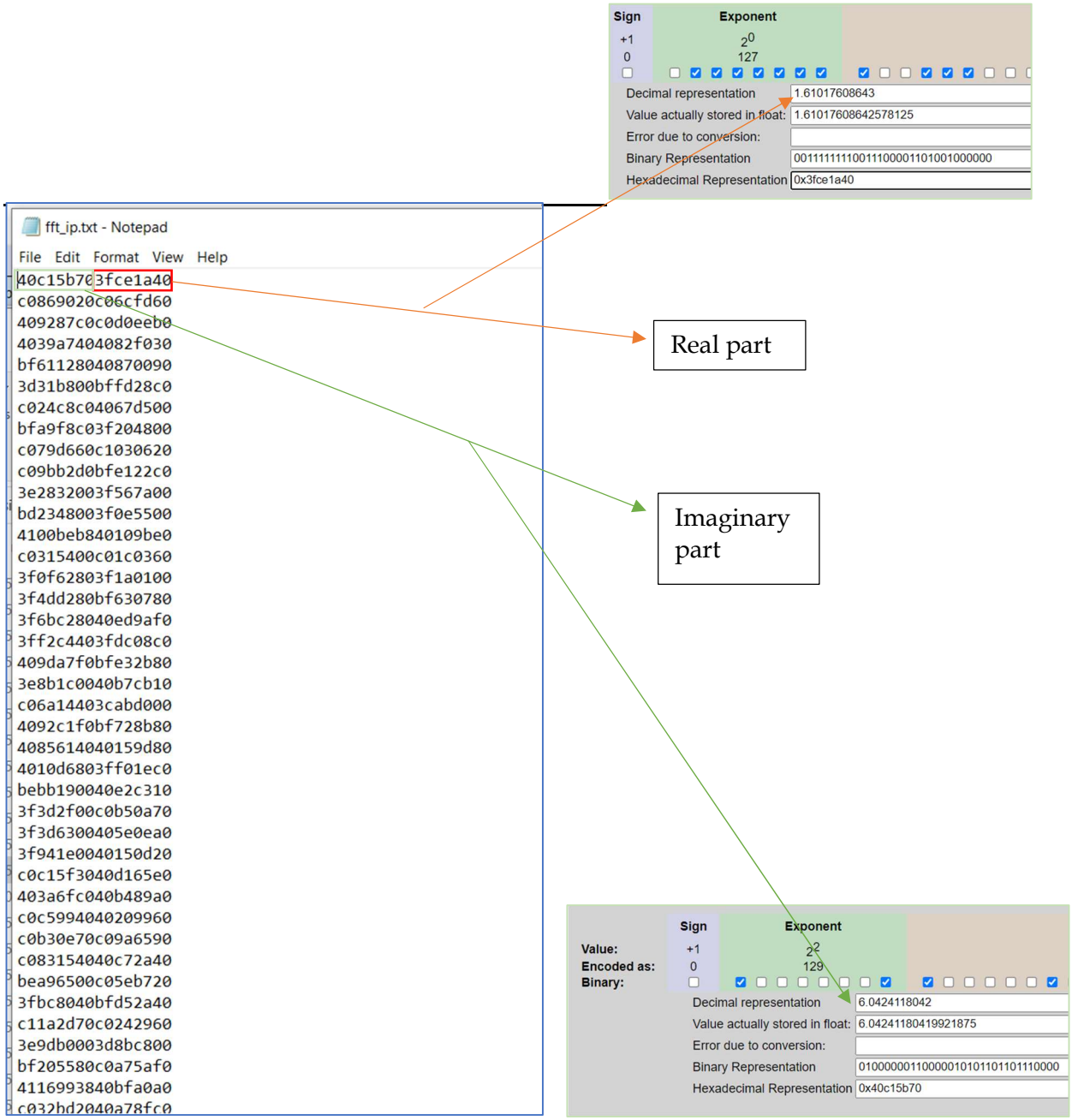


T_last (High) denoting the last sample of cp block



m_axis_data_tlast (High) indicating the last sample of FFT

FFT Samples : (From FFT IP)



Use below website to cross check the decimal format of real part(hex) and imaginary part(hex) separately with MATLAB samples below.

https://www.h-schmidt.net/FloatConverter/IEEE754.html

MATLAB CODE:

output_fft.txt | first.m

/MATLAB Drive/first.m

```
1   file1 = fopen('/MATLAB Drive/imag2.txt','r')
2   file2 = fopen('/MATLAB Drive/real2.txt','r')
3   file3=fopen('output_fft.txt','w')
4
5   %cp_added=[]
6   cp_removed=[]
7   c=0
8
9   while ~feof(file1) && ~feof(file2)
10      line1 = fgetl(file1);
11      line2 = fgetl(file2);
12      cp_removed = [cp_removed; complex(str2double(line2), str2double(line1))];
13  end
14
15  %cp_l = [320, 288];
16  %for i = 1:numel(cp_l)
17   %   for j = 1:4096
18    %      cp_removed = [cp_removed; cp_added(cp_l(i) + j + c)];
19    %  end
20     % c = c + 288;
21  %end
22
23  fft_result = fft(cp_removed);
24  for k = 1:numel(fft_result)
25
26  fprintf(file3, '%f +i%f\n', real(fft_result(k)), imag(fft_result(k)));
27  end
28
```

Command Window

FFT Samples : (From MATLAB)

output_fft.txt | first.m

/MATLAB Drive/output_fft.txt

```
1   1.610222 +i6.042513
2   -3.702850 +i-4.205078
3   -6.529000 +i4.579078
4   4.091888 +i2.900863
5   4.218861 +i-0.879127
6   -1.977777 +i0.043368
7   3.622489 +i-2.574772
8   0.626161 +i-1.327878
9   -8.188969 +i-3.903679
10  -1.758814 +i-4.865578
11  0.837893 +i0.164260
12  0.556022 +i-0.039839
13  2.259545 +i8.046584
14  -2.437674 +i-2.770792
15  0.601670 +i0.560103
16  -0.886743 +i0.804028
17  7.425178 +i0.921050
18  1.718929 +i1.896630
19  -1.774702 +i4.926623
20  5.743657 +i0.271717
21  0.021034 +i-3.657494
22  -0.947358 +i4.586230
23  2.337725 +i4.168176
24  1.875986 +i2.263073
25  7.086332 +i-0.365374
26  -5.657504 +i0.738987
27  3.469693 +i0.739791
28  2.328926 +i1.157157
```

Command Window

Real part

Imaginary part

Github:

https://github.com/velicharlagokulkumar/vitis-hls/tree/main/Assignment_8