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");  }} |

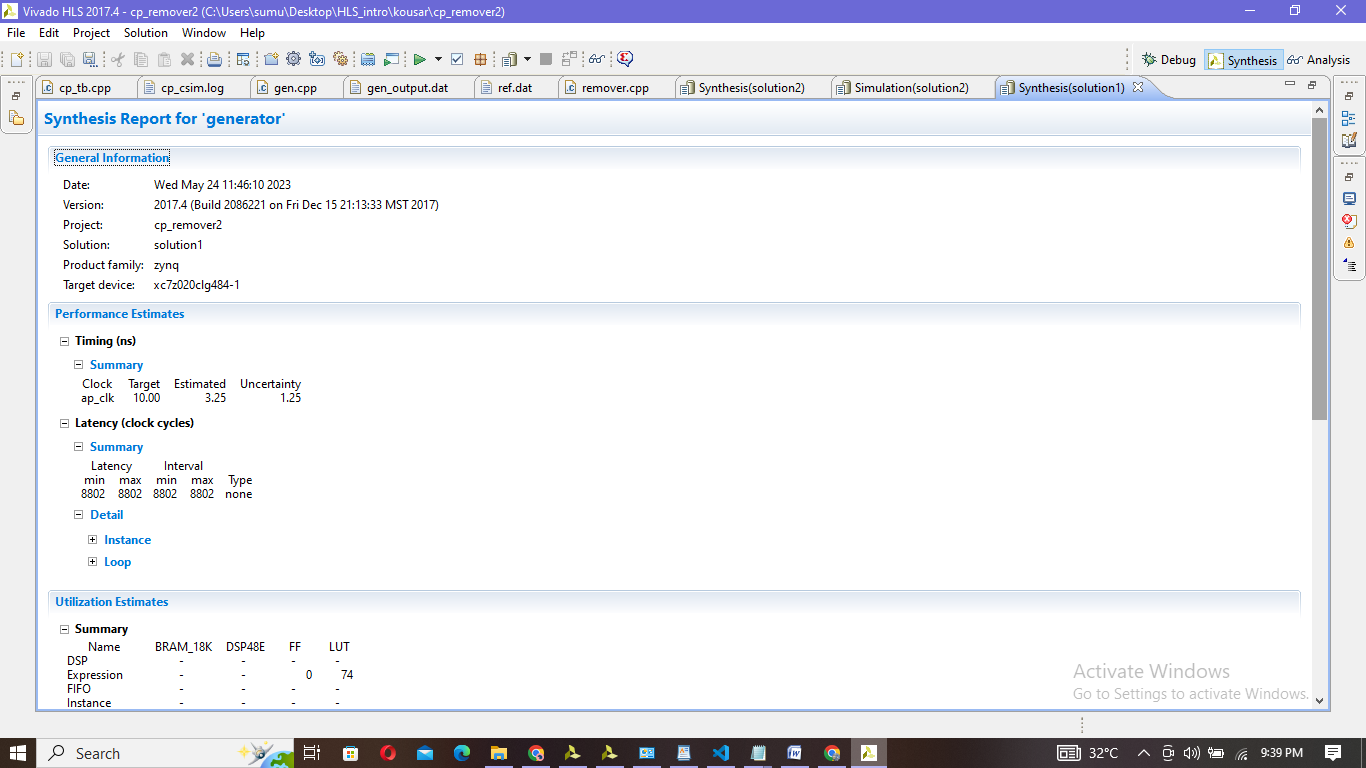
***FILES***

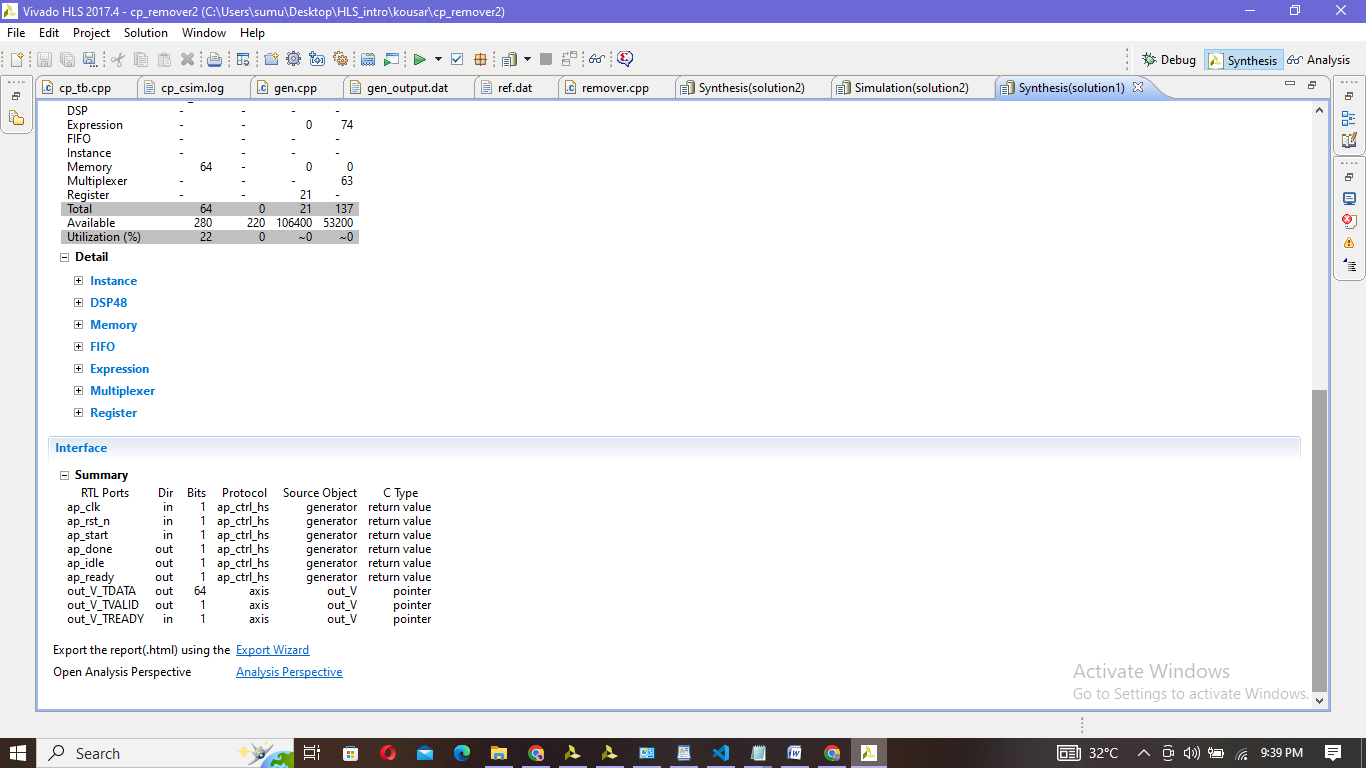
Input to cp\_remover:[https://github.com/kkousar/HLS/blob/main/hls%20files/gen\_output.dat](Assignment8.docx)

Output of cp\_remover: [https://github.com/kkousar/HLS/blob/main/hls%20files/cp\_output.dat](file:///C:\Users\sumu\Desktop\RTL-FPGA\Assignment8.docx)

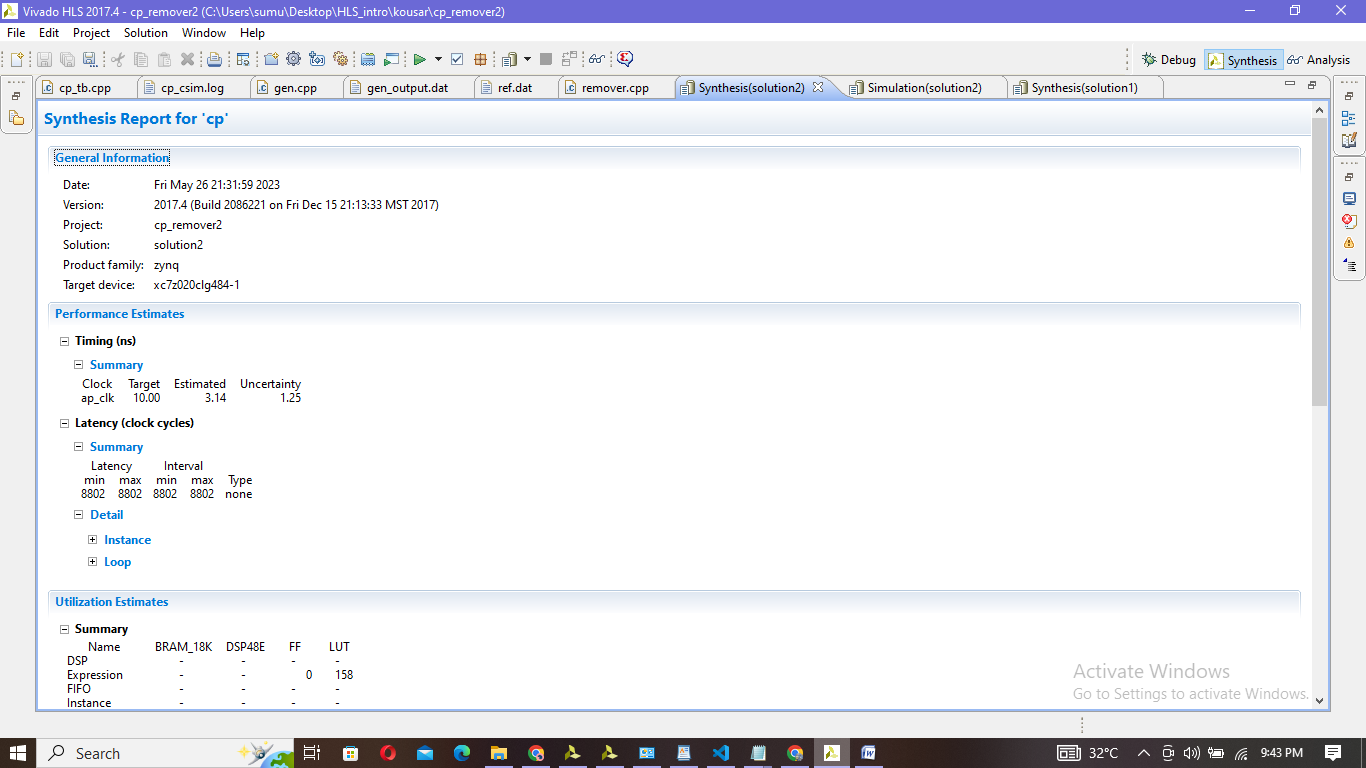
Ref input to testbench:[https://github.com/kkousar/HLS/blob/main/hls%20files/ref.dat](Assignment8.docx)

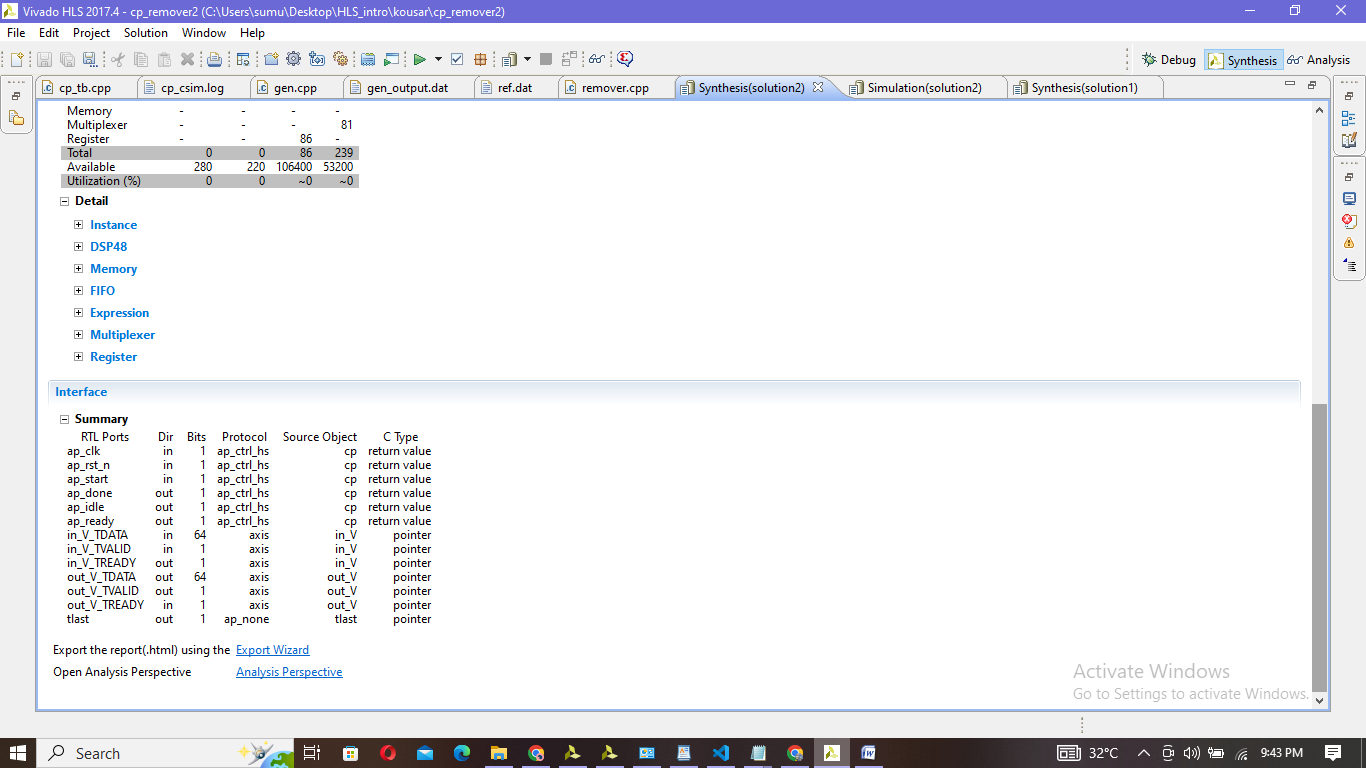
Generator Synthesis report:





Cp\_remover synthesis :



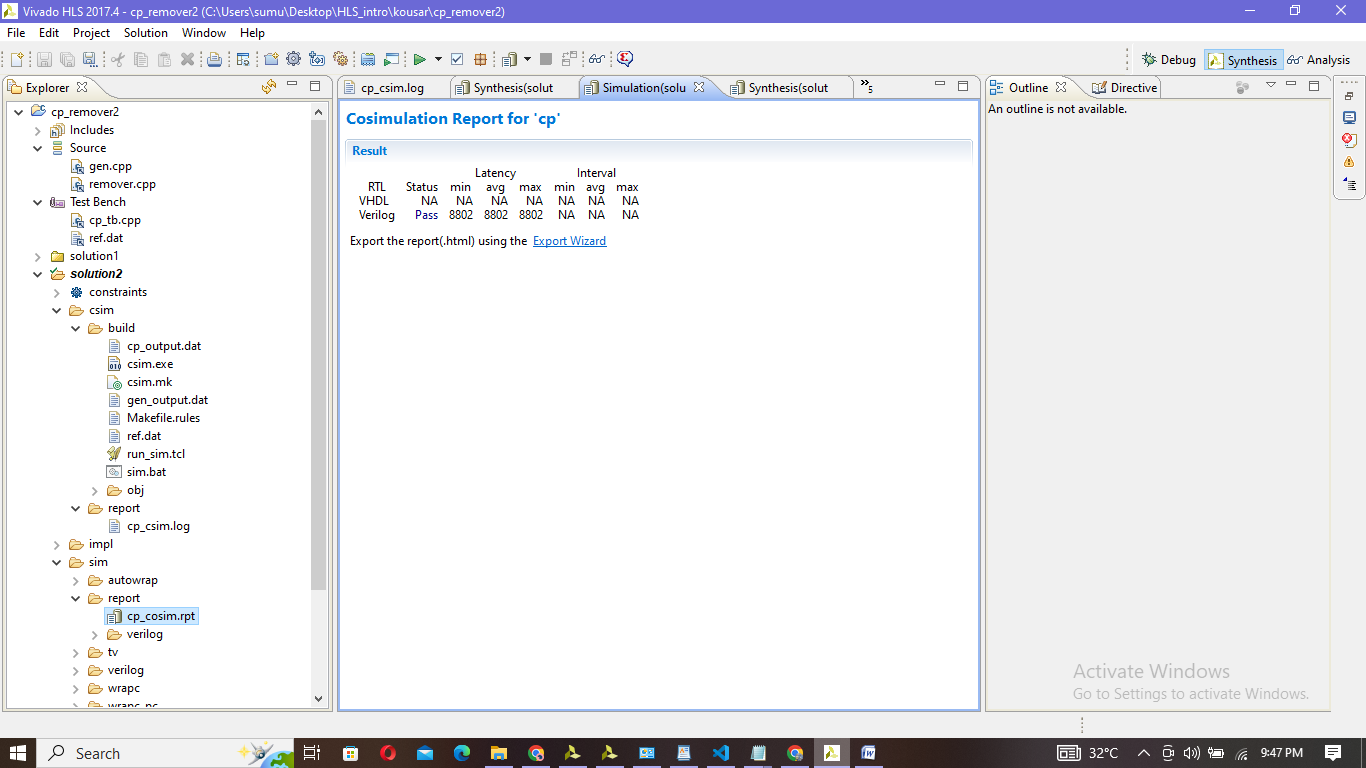


Simulation report

|  |
| --- |
| 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. |

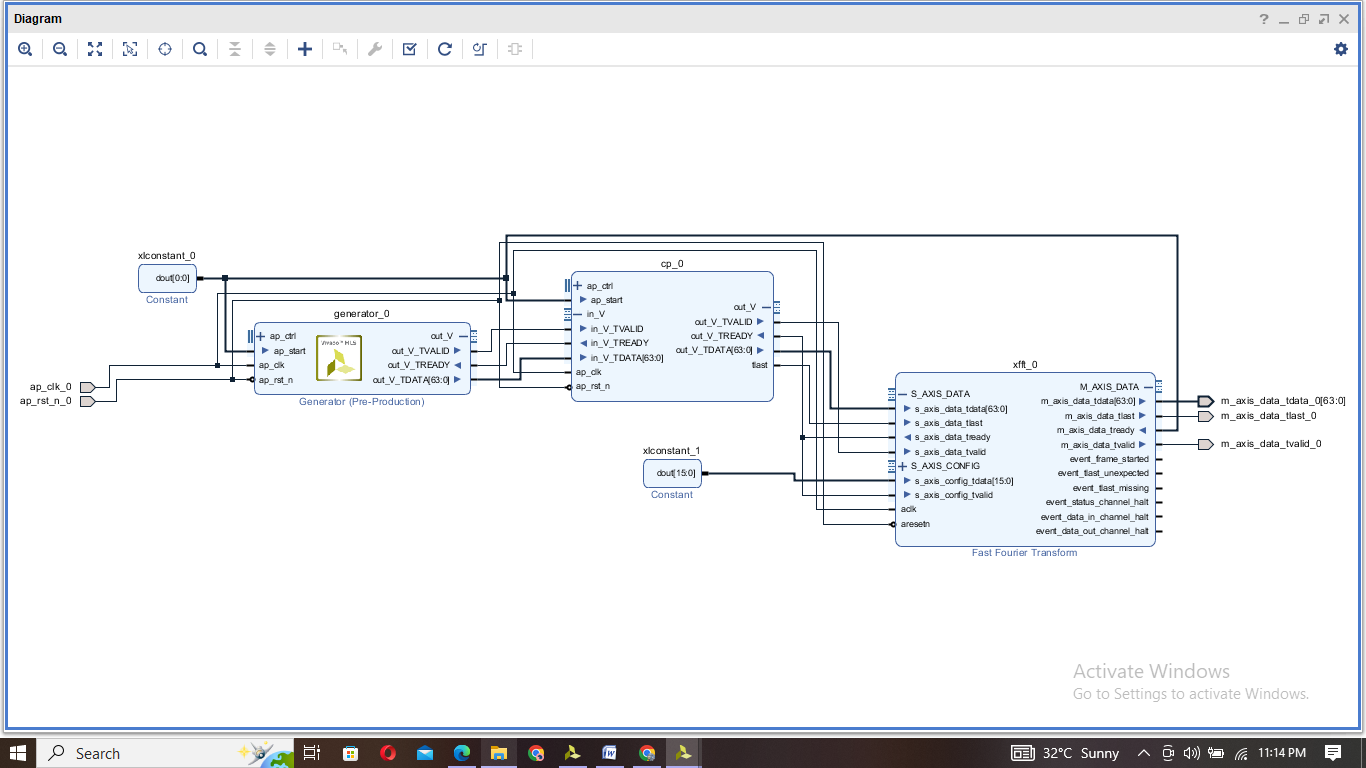
  
Matlab code:

|  |
| --- |
| 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](Assignment8.docx)

File3:-ouput of cp\_remover: [https://github.com/kkousar/HLS/blob/main/output\_cp.txt](Assignment8.docx)

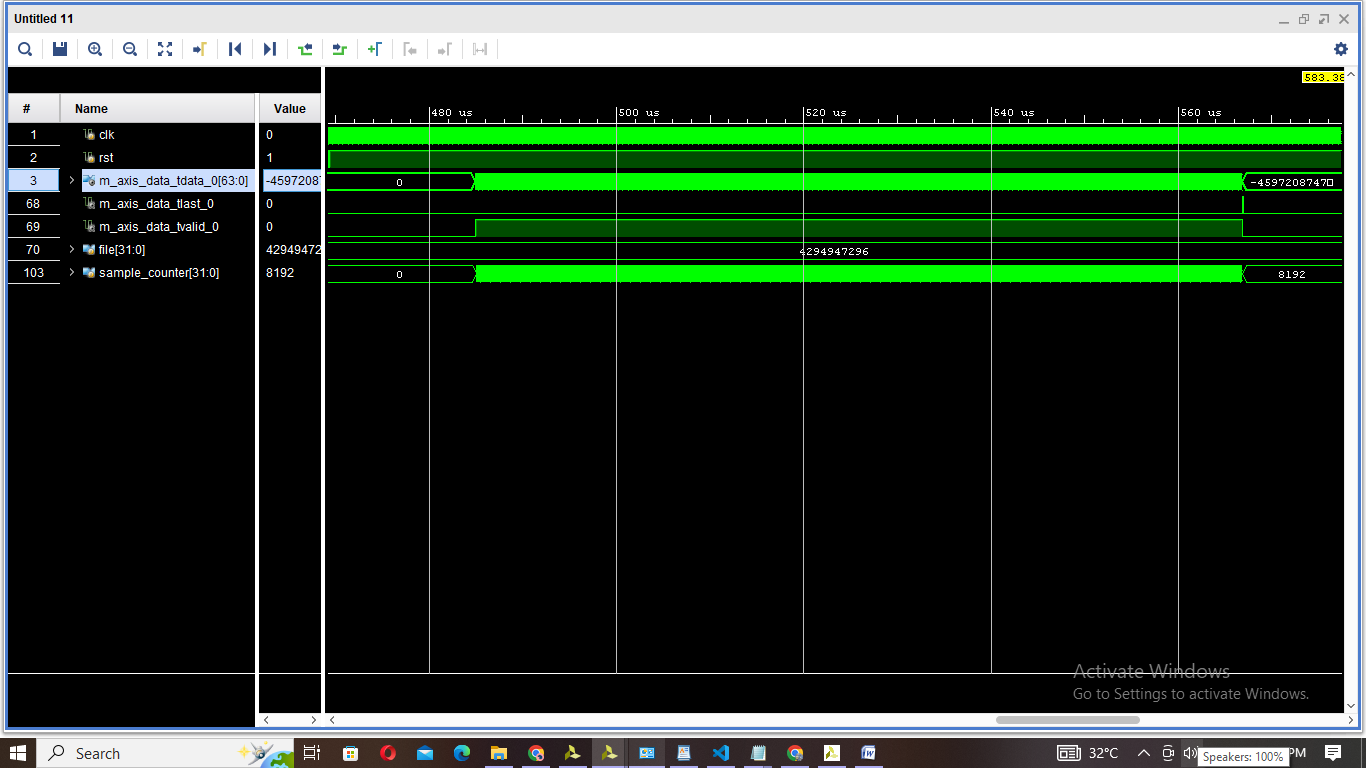
Vivado HLS:



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: vivado fft\_ip output:

|  |
| --- |
| Sample1:1.610222 +i6.042513🡪40c15c443fce1bc1  Sample2:-3.702850 +i-4.205078🡪c0869000c06cfb7f  …. |

|  |
| --- |
| 40c15ba03fce1980  c0869070c06cfd60 |

Vivado FFT IP output:[https://github.com/kkousar/HLS/blob/main/hls%20files/out\_fft.txt](Assignment8.docx)