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

Date:10-07-2023

Hls ASSIGNMENT-9

Zadoff-Chu Sequence Generator

Design a digital circuit using RTL and HLS for a Zadoff-Chu sequence generator.

Requirements

- Implement the following equation,
  - x(m) = e^ {-j * pi * m * (m+1) * u / L};
  - Where,
    - “u” is a sequence ID, possible values are 0 to 31
    - “L” is the length of the sequence, possible values are 139 and 839
  - Both these variables should be taken as an input.
- The input and output buses should use the AXIS interface.
- Implement a pipelined design.
- The implementation should use as minimum resources as possible.

Considerations

- A last signal should be used to indicate the end of the output sequence.
- The design implementation should be targeted on the ZCU111 FPGA board.

Deliverables

- Source code implementation of the design.
- Simulation results.
- Resource utilization report analysis.
- Timing report which contains the latency, interval, throughput and the Fmax of the circuit.
- Documentation detailing the design choices, and performance evaluation.

MATLAB:

SOURCE CODE:

Editor - C:\Users\velic\OneDrive\Desktop\vitis-hls\Assignment_9\MATLAB\zff_chu.m

zff_chu.m | real_matlab.txt | imag_matlab.txt

```
seq = zadoffChuSeq(25,139);
X=real(seq);
Y=imag(seq);
file1 = fopen('real_matlab.txt', 'w');
fprintf(file1, '%f\n', X);
fclose(file1);
%disp(X);
file2 = fopen('imag_matlab.txt', 'w');
fprintf(file2, '%f\n', Y);
fclose(file2);
disp(seq);
```

Output: (Zadoff- chu sequence)

Command Window

```
>> zff_chu
   1.0000 + 0.0000i
   0.4266 - 0.9044i
  -0.9693 + 0.2461i
   0.8789 - 0.4770i
   0.3004 + 0.9538i
  -0.3219 + 0.9468i
   0.1687 + 0.9857i
   0.9746 - 0.2241i
  -0.9875 - 0.1576i
   0.8323 - 0.5544i
   0.7788 + 0.6273i
   0.6869 + 0.7268i
   0.9837 - 0.1798i
  -0.6702 - 0.7421i
   0.7497 + 0.6618i
  -0.8679 + 0.4967i
  -0.9693 - 0.2461i
  -0.9936 + 0.1128i
   0.0339 + 0.9994i
   0.4670 - 0.8842i
   0.1240 + 0.9923i
  -0.9571 + 0.2896i
  -0.9997 + 0.0226i
  -0.6360 + 0.7717i
   0.9634 + 0.2679i
  -0.9571 - 0.2896i
```

HLS:

Design:

Main function: (Sequence Generator)

zff_chu.cpp | zff_chu_tb.cpp | _csim.log | Synthesis Summary(solution1)

```cpp
#include <hls_stream.h>
#include <math.h>
#include<complex>
using namespace std;

typedef complex <float> data_stream;
void zadoff_chu_generator_hls(hls::stream<data_stream>& out_stream, int length, int u,hls::stream<bool> &tlast) {
#pragma HLS INTERFACE mode=axis register_mode=off port=tlast
#pragma HLS INTERFACE mode=axis register_mode=off port=out_stream
   data_stream out_data;

    for (int m = 0; m < length; m++) {
        #pragma HLS PIPELINE II=1
        float angle = M_PI * m * (m + 1) * u / length;
        float real = cos(angle);
        float imag = -sin(angle);
        out_stream.write(complex<float>(real,imag));
    }
    tlast.write(1);
}
```

Self-checking Test Bench:

hls_zff_seque... | zff_chu_tb.cpp | Synthesis Su... | Co-simulatio... | real_matlab.txt | Synthesis Det... | Synthesis Det... | Sy

```cpp
#include <iostream>
#include <hls_stream.h>
#include <math.h>
#include <complex>
using namespace std;
#include <fstream>

typedef complex <float> data_stream;

void zadoff_chu_generator_hls(hls::stream<data_stream>& out_stream, int length, int u,hls::stream<bool> &tlast);
int main() {
    const int LENGTH = 139;
    const int U = 25;

    double arr1[LENGTH];
    double arr2[LENGTH];
    data_stream arr3[LENGTH];
    int threshold = 5;
    int flag=0;
    hls::stream<data_stream> out_stream;
    hls::stream<bool> t_last;

    ifstream File1("real_matlab.txt");
    for (int i = 0; i < LENGTH; i++){
    File1>>arr1[i];
    }
    File1.close();

    ifstream File2("imag_matlab.txt");
    for (int i = 0; i < LENGTH; i++){
    File2>>arr2[i];
    }
   File2.close();

    zadoff_chu_generator_hls(out_stream, LENGTH, U,t_last);
```

```
36
37      ofstream File5("hls_zff_sequence.txt");
38      for (int i = 0; i < LENGTH; i++){
39          arr3[i] = out_stream.read();
40          File5 << arr3[i]<<endl;
41        }
42      File5.close();
43
44      ofstream File3("generator_real.txt");
45      for (int i = 0; i < LENGTH; i++){
46          File3<<arr3[i].real()<<endl;
47        }
48        File3.close();
49
50        ofstream File4("generator_imag.txt");
51           for (int i = 0; i < LENGTH; i++){
52              File4<<arr3[i].imag()<<endl;
53             }
54             File4.close();
55
56             for (int i = 0; i < LENGTH; i++) {
57                 float absoluteDiff = abs(arr1[i] - arr3[i].real());
58                 float thresholdValue = (threshold * abs(arr1[i])) / 100;
59                 if (absoluteDiff > thresholdValue) {
60                     cout << "Sample " << i << " exceeds the threshold: " << arr1[i] << ", " << arr3[i].real()<< endl;
61                     flag=1;
62                 }
63
64             }
65  if(flag==1)
66  {
67      cout<<"Few Samples are exceeded the threshold when compared with MATLAB reference output"<<endl;
68  }
69  else
70  {
71      cout<<"All Samples are within the tolerance band when compared with MATLAB reference output"<<endl;
72  }
73
74      return 0;
75  }
76
```

Input Files: (golden data from MATLAB)

Real Values of Sequence:

Editor - C:\Users\velic\OneDrive\Desktop\vitis-hls\Assignment_9\MATLAB\real_matlab.txt

zff_chu.m | real_matlab.txt | imag_matlab.txt | +

```
1.000000
0.426597
-0.969254
0.878907
0.300406
-0.321885
0.168700
0.974567
-0.987511
0.832254
0.778803
0.686850
0.983698
-0.670250
0.749663
-0.867903
-0.969254
-0.993622
0.033896
0.467031
0.123988
-0.957145
-0.999745
-0.636030
```

Imaginary Values of Sequence:

Editor - C:\Users\velic\OneDrive\Desktop\vitis-hls\Assignment_9\MATLAB\imag_matlab.txt

zff_chu.m | real_matlab.txt | imag_matlab.txt

```
0.000000
-0.904442
0.246062
-0.476993
0.953811
0.946779
0.985667
-0.224095
-0.157551
-0.554395
0.627269
0.726799
-0.179827
-0.742136
0.661820
0.496734
-0.246062
0.112767
0.999425
-0.884241
0.992284
0.289609
0.022599
0.771665
```

Output File: (data returned by function)

zff_chu_tb.cpp | hls_zff_seque... | Synthesis Su... | Co-simulatio...

```
(1,-0)
(0.426597,-0.904442)
(-0.969254,0.246062)
(0.878907,-0.476993)
(0.300406,0.953811)
(-0.321885,0.946779)
(0.168699,0.985668)
(0.974567,-0.224095)
(-0.987511,-0.15755)
(0.832253,-0.554395)
(0.778803,0.627268)
(0.686849,0.7268)
(0.983698,-0.179831)
(-0.670248,-0.742137)
(0.749665,0.661818)
(-0.8679,0.49674)
(-0.969254,-0.246064)
(-0.993621,0.112768)
(0.0338971,0.999425)
(0.467032,-0.88424)
(0.123995,0.992283)
(-0.957142,0.28962)
(-0.999745,0.0225983)
(-0.63602,0.771673)
(0.963444,0.267911)
(-0.957145,-0.289609)
(0.686848,-0.726801)
(0.995916,0.0902791)
(0.990819,-0.135193)
(0.0790286,-0.996872)
(-0.670239,0.742145)
(0.257028,-0.966404)
(0.974564,0.22411)
```

C simulation printed output:

zff_chu_tb.cpp | _csim.log | Synthesis Summary(solution1) | Co-simulation Report(solution1) | real_matlab.txt

```
INFO: [SIM 2] *************** CSIM start ***************
INFO: [SIM 4] CSIM will launch GCC as the compiler.
   Compiling ../../../Sources/zff_chu_tb.cpp in debug mode
   Compiling ../../../Sources/zff_chu.cpp in debug mode
   Generating csim.exe
All Samples are within the tolerance band when compared with MATLAB 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 139
INFO: [SIM 1] CSim done with 0 errors.
INFO: [SIM 3] *************** CSIM finish ***************

```

Synthesis Report:

zff_chu_tb.cpp | _csim.log | Synthesis Summary(solu... | Co-simulation Report(so... | real_matlab.txt | Synthesis Details(solutio... | Synthesis Details(solutio... | Synthesis Details(solutio...

project_39/testbench/zff_chu_tb.cpp

Synthesis Summary Report of 'zadoff_chu_generator_hls'

General Information

Date:	Sat Jul 1 08:58:06 2023	Solution:	solution1 (Vivado IP Flow Target)
Version:	2022.2 (Build 3670227 on Oct 13 2022)	Product family:	zynquplus
Project:	project_39	Target device:	xczu7ev-ffvc1156-2-e

Timing Estimate

Target	Estimated	Uncertainty
10.00 ns	7.042 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	UR
zadoff_chu_generator_hls_Pipeline_VITIS_LOOP_12_1				-	-	-	-	-	-	no	0	57	4596	6079	
VITIS_LOOP_12_1				-	-	-	53	1	-	yes	-	-	-	-	
sin_or_cos_float_s				-	10	100.000	-	1	-	yes	0	12	1541	2649	
sin_or_cos_float_s				-	10	100.000	-	1	-	yes	0	12	1541	2649	

Performance Pragma

Modules & Loops	Target TI(cycles)	TI(cycles)	TI met
zadoff_chu_generator_hls	-	-	-
zadoff_chu_generator_hls_Pipeline_VITIS_LOOP_12_1	-	-	-
VITIS_LOOP_12_1	-	-	-
sin_or_cos_float_s	-	-	-
sin_or_cos_float_s	-	-	-

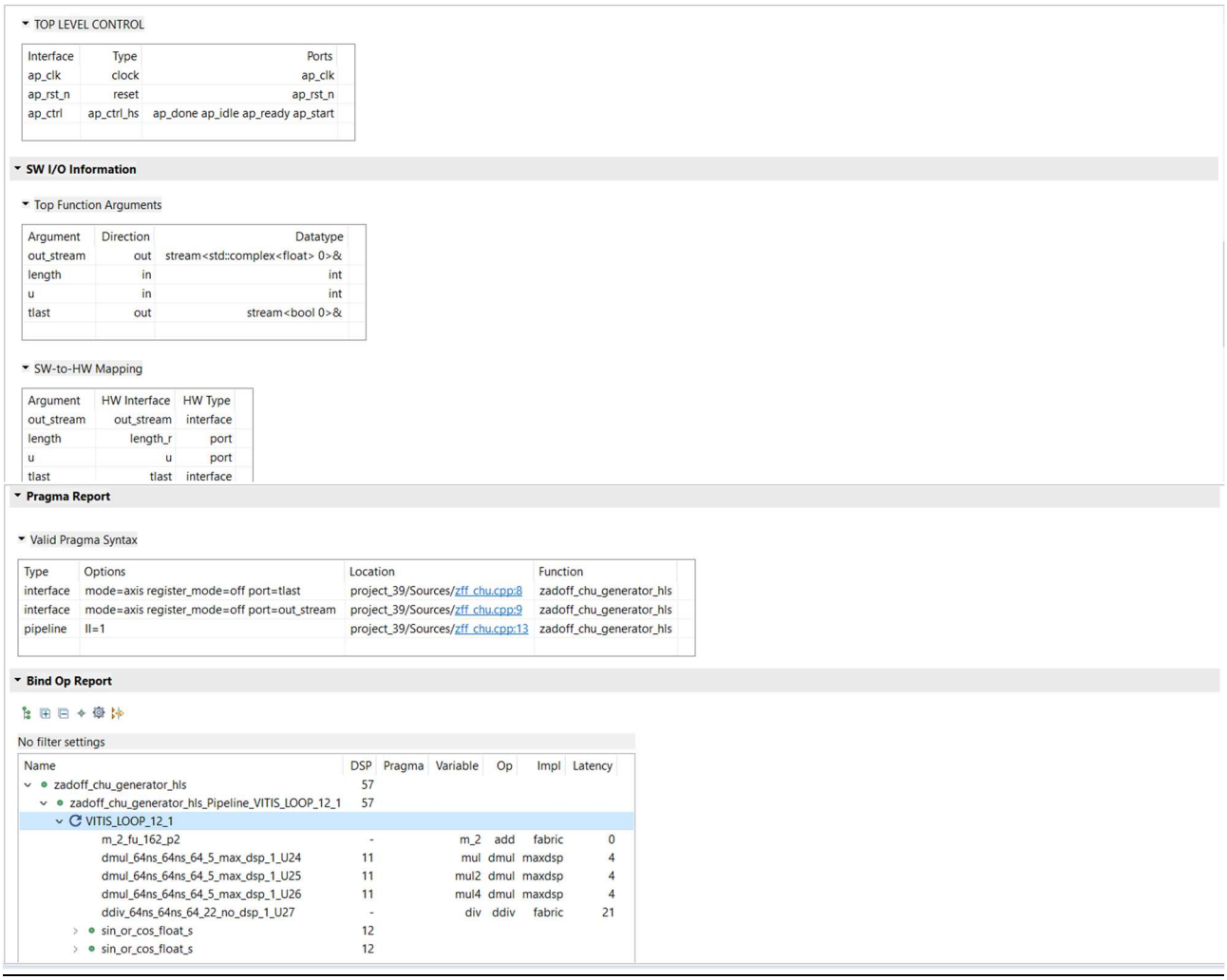
HW Interfaces

AXIS

Interface	Register Mode	TDATA	TREADY	TVALID
out_stream	off	64	1	1
tlast	off	8	1	1

Other Ports

Interface	Mode	Bitwidth
length_r	ap_none	32
u	ap_none	32



▾ 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
out_stream	out	stream<std::complex<float> 0>&
length	in	int
u	in	int
tlast	out	stream<bool 0>&

▾ SW-to-HW Mapping

Argument	HW Interface	HW Type
out_stream	out_stream	interface
length	length_r	port
u	u	port
tlast	tlast	interface

▾ Pragma Report

▾ Valid Pragma Syntax

Type	Options	Location	Function
interface	mode=axis register_mode=off port=tlast	project_39/Sources/zff_chu.cpp:8	zadoff_chu_generator_hls
interface	mode=axis register_mode=off port=out_stream	project_39/Sources/zff_chu.cpp:9	zadoff_chu_generator_hls
pipeline	II=1	project_39/Sources/zff_chu.cpp:13	zadoff_chu_generator_hls

▾ Bind Op Report

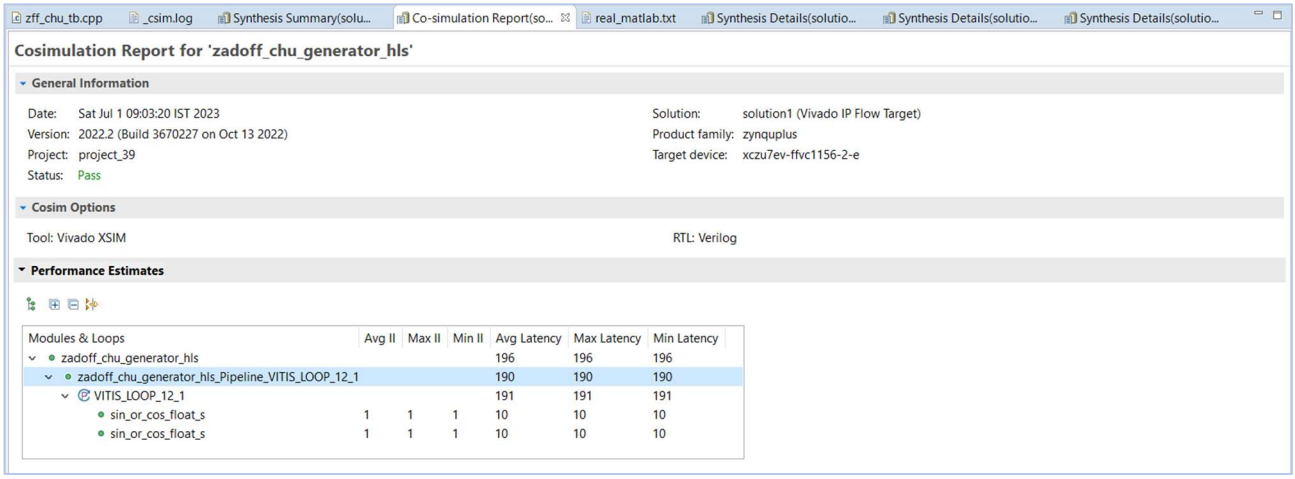
No filter settings

Name	DSP	Pragma	Variable	Op	Impl	Latency
zadoff_chu_generator_hls	57					
zadoff_chu_generator_hls_Pipeline_VITIS_LOOP_12_1	57					
VITIS_LOOP_12_1						
m_2_fu_162_p2	-		m_2	add	fabric	0
dmul_64ns_64ns_64_5_max_dsp_1_U24	11		mul	dmul	maxdsp	4
dmul_64ns_64ns_64_5_max_dsp_1_U25	11		mul2	dmul	maxdsp	4
dmul_64ns_64ns_64_5_max_dsp_1_U26	11		mul4	dmul	maxdsp	4
ddiv_64ns_64ns_64_22_no_dsp_1_U27	-		div	ddiv	fabric	21
sin_or_cos_float_s	12					
sin_or_cos_float_s	12					

Co-simulation printed output:

```
////////////////////////////////////////////////////////////////////////////////
// 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 [n/a] @ "125000"
// RTL Simulation : 1 / 1 [n/a] @ "2105000"
////////////////////////////////////////////////////////////////////////////////
$finish called at time : 2165 ns : File "C:/Users/velic/OneDrive/Desktop/vitis-hls/Assignment_9/HLS/project_39/solution1/sim/verilog/zadoff_chu_generator_hls.autotb.v" Li
## quit
INFO: [Common 17-206] Exiting xsim at Sat Jul  1 13:03:15 2023...
INFO: [COSIM 212-316] Starting C post checking ...
All Samples are within the tolerance band when compared with MATLAB 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 139
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 calcul
INFO: [HLS 200-111] Finished Command cosim_design CPU user time: 2 seconds. CPU system time: 1 seconds. Elapsed time: 229.744 seconds; current allocated memory: 8.480 MB.
INFO: [HLS 200-112] Total CPU user time: 5 seconds. Total CPU system time: 3 seconds. Total elapsed time: 243.895 seconds; peak allocated memory: 113.258 MB.
Finished C/RTL cosimulation.
```

Co-simulation Report:



zff_chu_tb.cpp | _csim.log | Synthesis Summary(solu... | Co-simulation Report(so... | real_matlab.txt | Synthesis Details(solutio... | Synthesis Details(solutio... | Synthesis Details(solutio...

Cosimulation Report for 'zadoff_chu_generator_hls'

General Information

Date: Sat Jul 1 09:03:20 IST 2023
Version: 2022.2 (Build 3670227 on Oct 13 2022)
Project: project_39
Status: Pass

Solution: solution1 (Vivado IP Flow Target)
Product family: zynquplus
Target device: xczu7ev-ffvc1156-2-e

Cosim Options

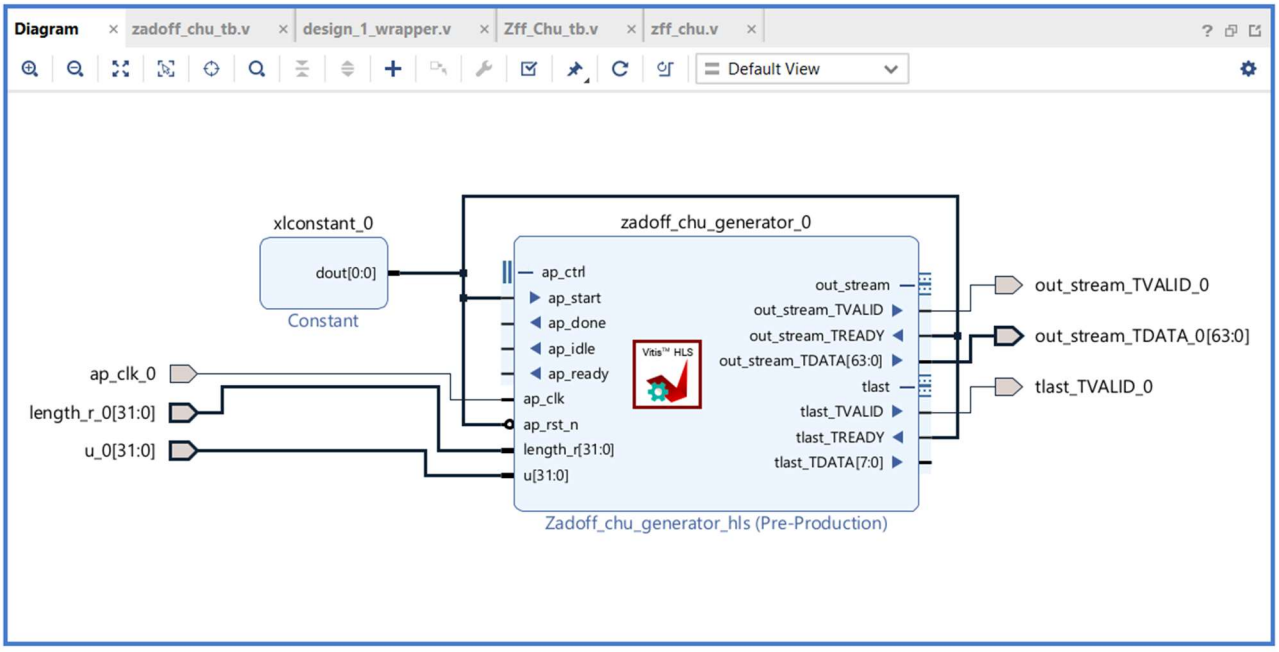
Tool: Vivado XSIM
RTL: Verilog

Performance Estimates

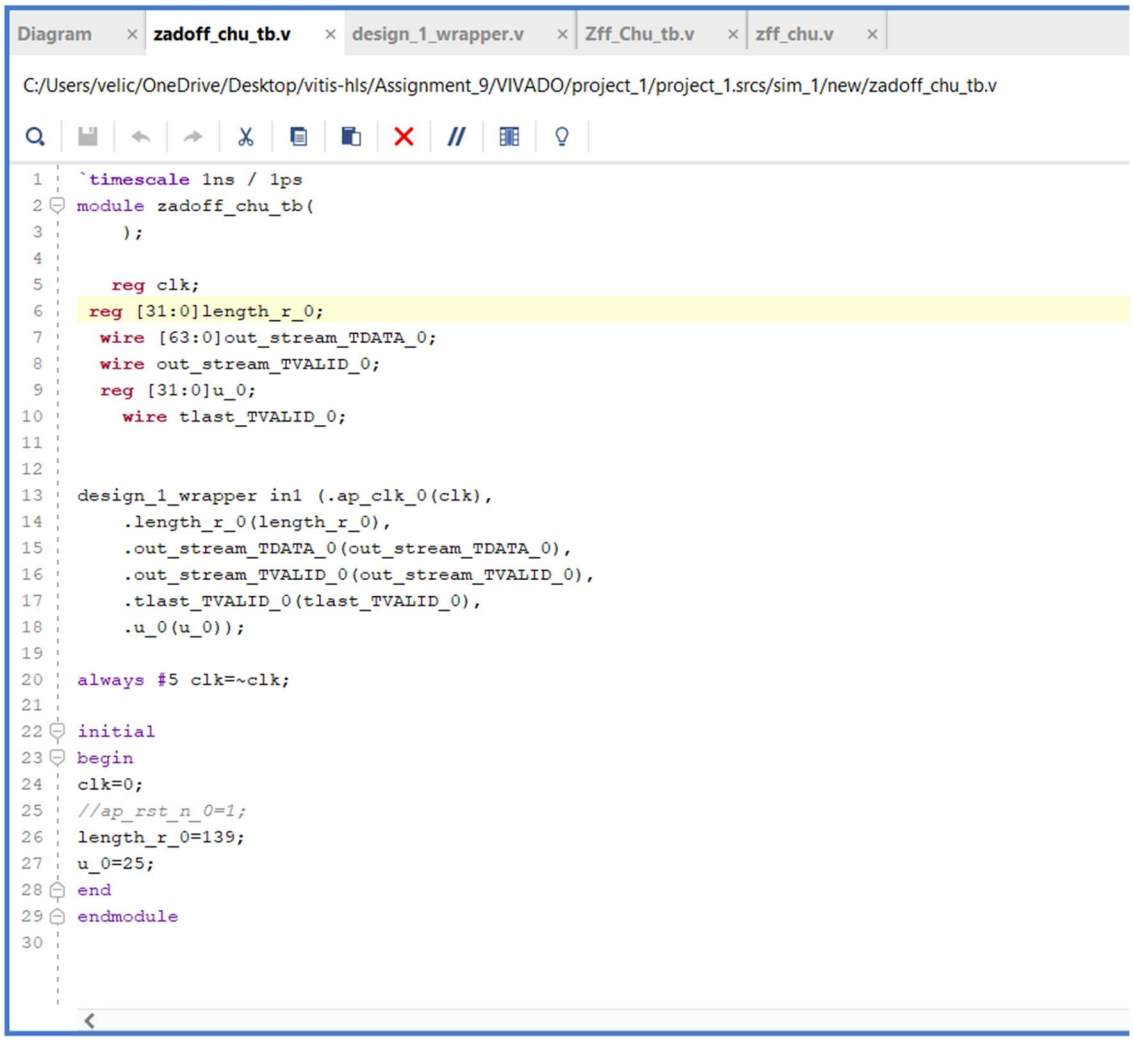
Modules & Loops	Avg II	Max II	Min II	Avg Latency	Max Latency	Min Latency
zadoff_chu_generator_hls				196	196	196
zadoff_chu_generator_hls_Pipeline_VITIS_LOOP_12_1				190	190	190
VITIS_LOOP_12_1				191	191	191
sin_or_cos_float_s	1	1	1	10	10	10
sin_or_cos_float_s	1	1	1	10	10	10

VIVADO:

Block Design: (Vitis IP)

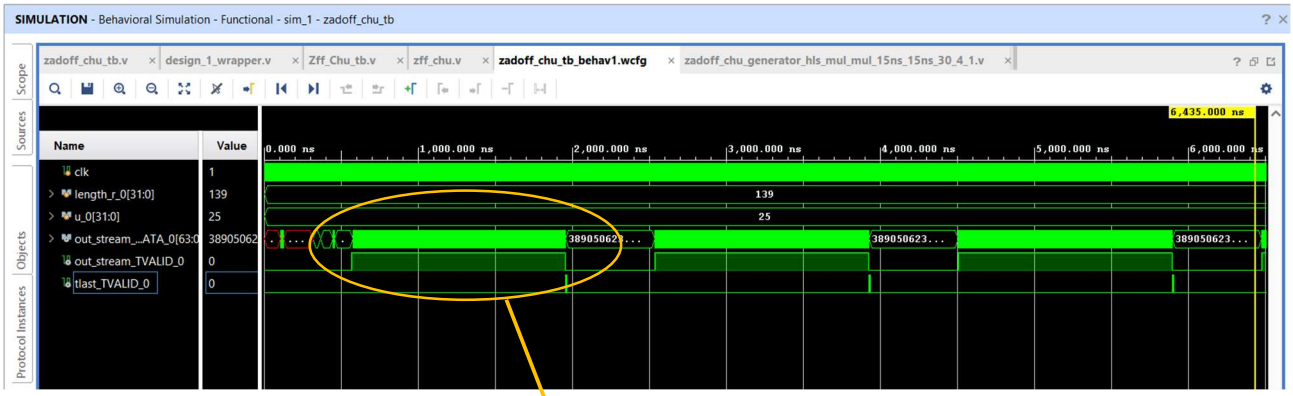


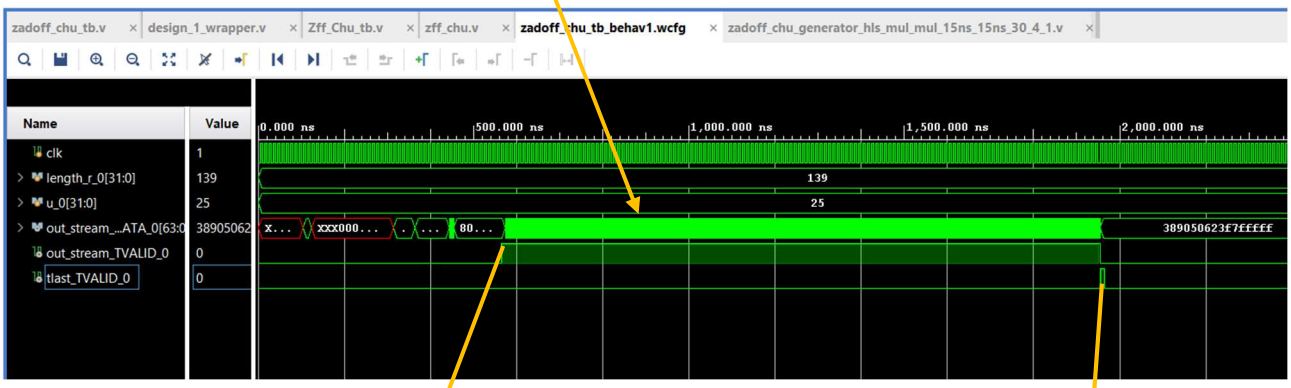
Test Bench:

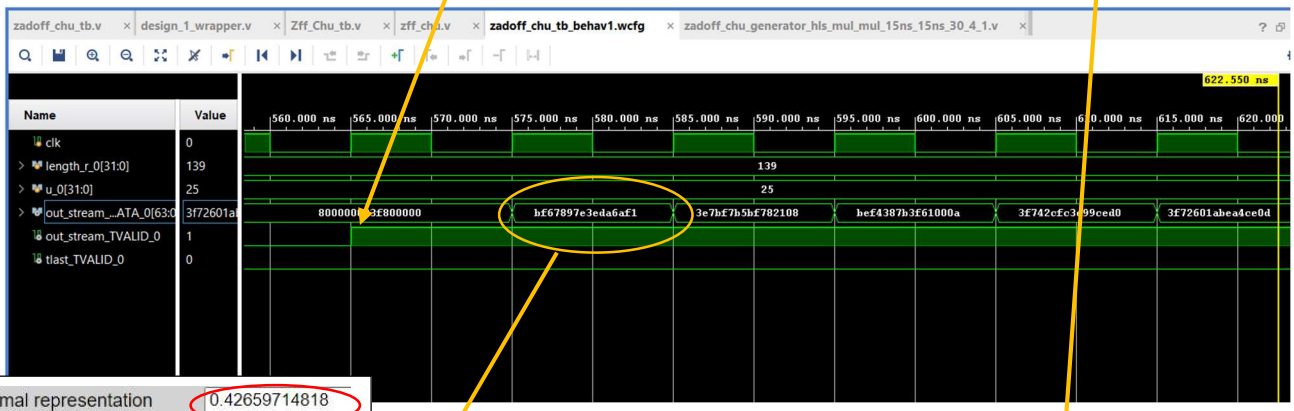


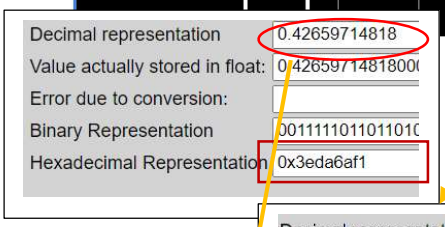

```
1  `timescale 1ns / 1ps
2  module zadoff_chu_tb(
3      );
4
5     reg clk;
6   reg [31:0]length_r_0;
7    wire [63:0]out_stream_TDATA_0;
8    wire out_stream_TVALID_0;
9    reg [31:0]u_0;
10     wire tlast_TVALID_0;
11
12
13 design_1_wrapper in1 (.ap_clk_0(clk),
14     .length_r_0(length_r_0),
15     .out_stream_TDATA_0(out_stream_TDATA_0),
16     .out_stream_TVALID_0(out_stream_TVALID_0),
17     .tlast_TVALID_0(tlast_TVALID_0),
18     .u_0(u_0));
19
20 always #5 clk=~clk;
21
22 initial
23 begin
24 clk=0;
25 //ap_rst_n_0=1;
26 length_r_0=139;
27 u_0=25;
28 end
29 endmodule
30
```

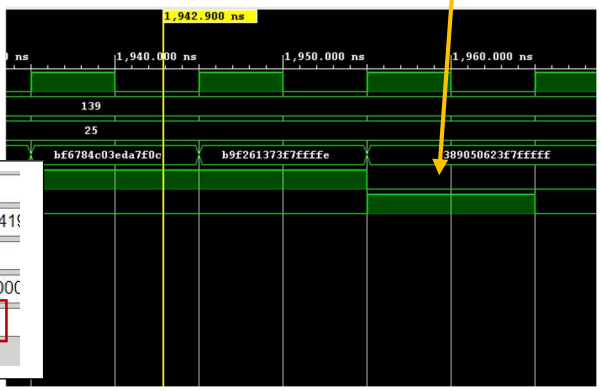
Simulation Window:

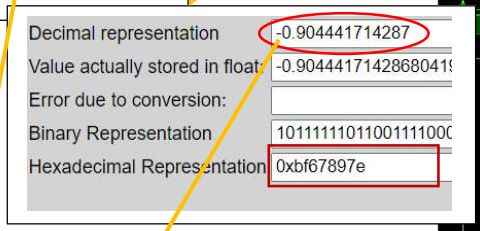




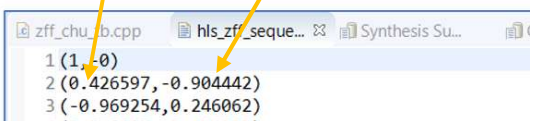








Output File: (data returned by function)

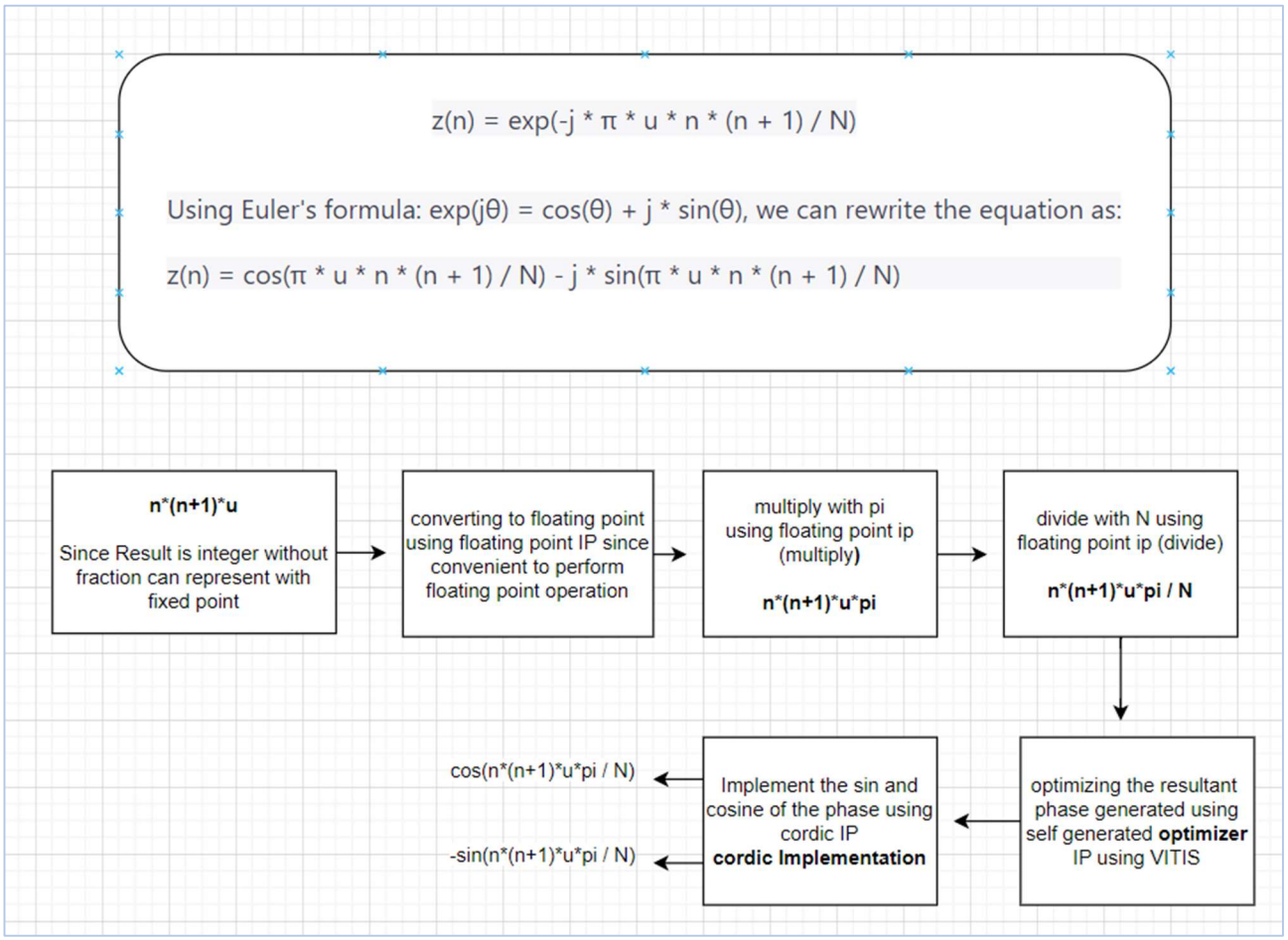


https://www.h-schmidt.net/FloatConverter/IEEE754.html

(In Verilog)

Ideology:

Zadoff -chu Sequence generator



Source:

https://github.com/velicharlagokulkumar/vitis-hls/tree/main/Assignment_9/VIVADO/project_2/project_2.srcs/sources_1/new/zff_2.v

Vitis Optimizer:

Ideology:

The input angle, PHASE_IN was given to the Cordic IP in **2**QN format with integer width of 3 (along with sign bit) that means ,I can express numbers in the range of -**4**.0.... to **3**.992.. for feeding my IP with an angle for producing sin and cosine of that angle provided

..as from the reference of pg105,Vivado IDE

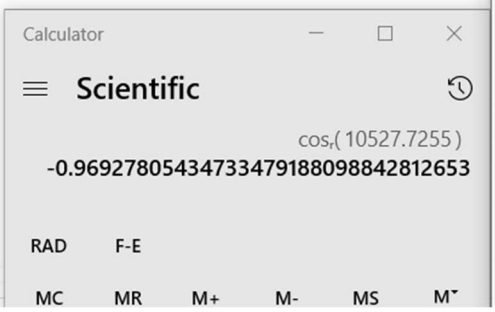
But the problem here is my angle was reaching to almost around **10527**.7.... (integer part **bold**) which i can not feed to my IP as it is expecting some value from the above mentioned range.

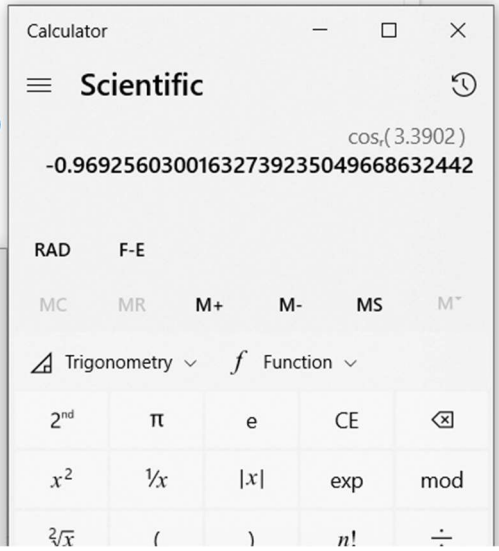
Let
my angle =10527.72558539568

Quotient= angle/2*pi ==> 1675.5395664 (Taking only integer part)
Quotient2=Quotient*2*pi ==> 1675*2*pi=10524.3353
reminder=angle-Quotinet2 ==>10527.7255-10524.3353=3.3902

reminder=3.3902 (my optimized phase)

cos(3.3902)=-0.969
cos(10527.7255)=-0.96925





Source: (optimizer IP)

opti.cpp | opti_tb.cpp | Co-simulation Report(solution1) | phase_optimizer_cordic_csim.log | Synthesis Summary(solution1)

```
#include <hls_stream.h>
#include <math.h>
#include<complex>
using namespace std;
#include<ap_fixed.h>

typedef ap_fixed <32,16> in_stream;
typedef ap_fixed <16,3> out_stream;

void phase_optimizer_cordic(hls::stream<out_stream>& outstream,hls::stream<in_stream>& instream) {
#pragma HLS PIPELINE II=1
#pragma HLS INTERFACE mode=axis register_mode=off port=outstream
#pragma HLS INTERFACE mode=axis register_mode=off port=instream

    float quotient;
    float PI = 3.14159265358979;
    float reminder;
    float quotient2;
    float angle= instream.read();
      quotient =round(angle/(2*PI));
      quotient2=quotient*(2*PI);
      reminder=angle-quotient2;
      outstream.write(reminder);

}

```

Test Bench:

opti.cpp | opti_tb.cpp | Co-simulation Report(solution1) | phase_optimizer_cordic_csim.log | Synthesis Summary(solu

```
#include <hls_stream.h>
#include <math.h>
#include<complex>
using namespace std;
#include<ap_fixed.h>

typedef ap_fixed <32,16> in_stream;
typedef ap_fixed <16,3> out_stream;

void phase_optimizer_cordic(hls::stream<out_stream>& outstream,hls::stream<in_stream>& instream);
int main()
{
    hls::stream<out_stream> outstream;
    hls::stream<in_stream> instream;

    instream.write(11.3006896972656);
    phase_optimizer_cordic(outstream,instream);
    cout<<outstream.read()<<endl;
    return 0;
}

```

C-Simulation:

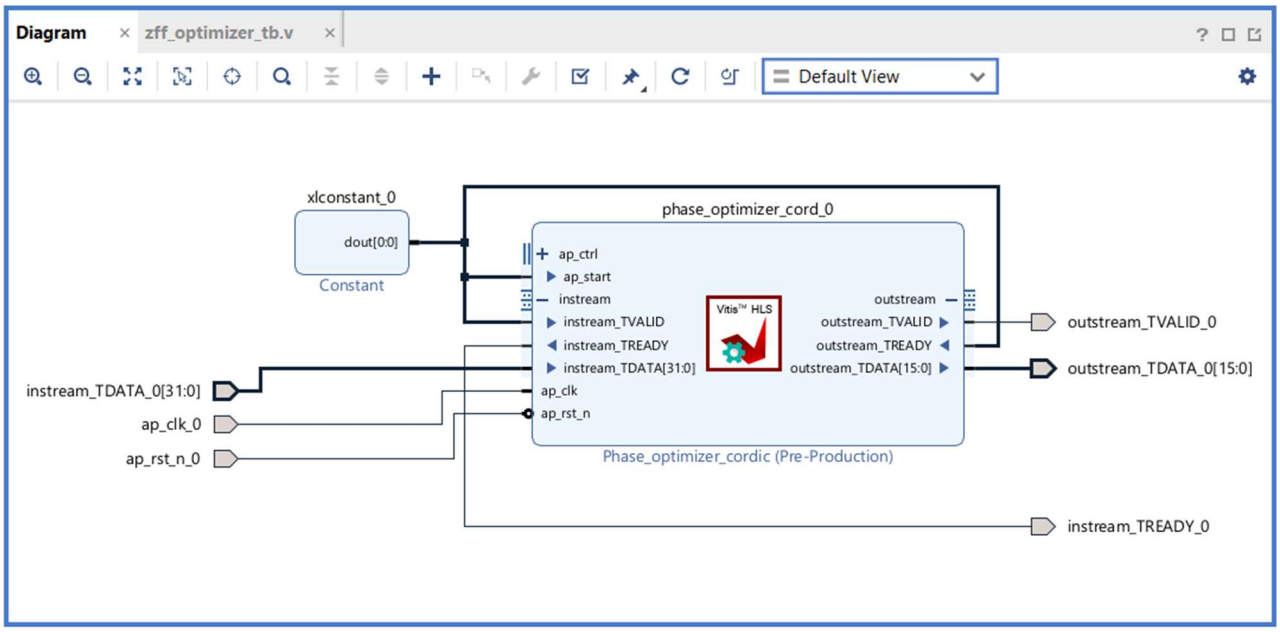
opti.cpp | opti_tb.cpp | Co-simulation Report(solution1) | phase_optimizer_cordic_csim.log | Synthesis Summ

```
INFO: [SIM 2] *************** CSIM start ***************
INFO: [SIM 4] CSIM will launch GCC as the compiler.
   Compiling ../../../../opti_tb.cpp in debug mode
   Compiling ../../../../opti.cpp in debug mode
   Generating csim.exe
-1.26575
INFO [HLS SIM]: The maximum depth reached by any hls::stream() instance in the design is 1
INFO: [SIM 1] CSim done with 0 errors.
INFO: [SIM 3] *************** CSIM finish ***************

```

Testing optimizer in vivado:

Block Design:



Test Bench:

zff_optimizer_tb.v | Untitled 3*

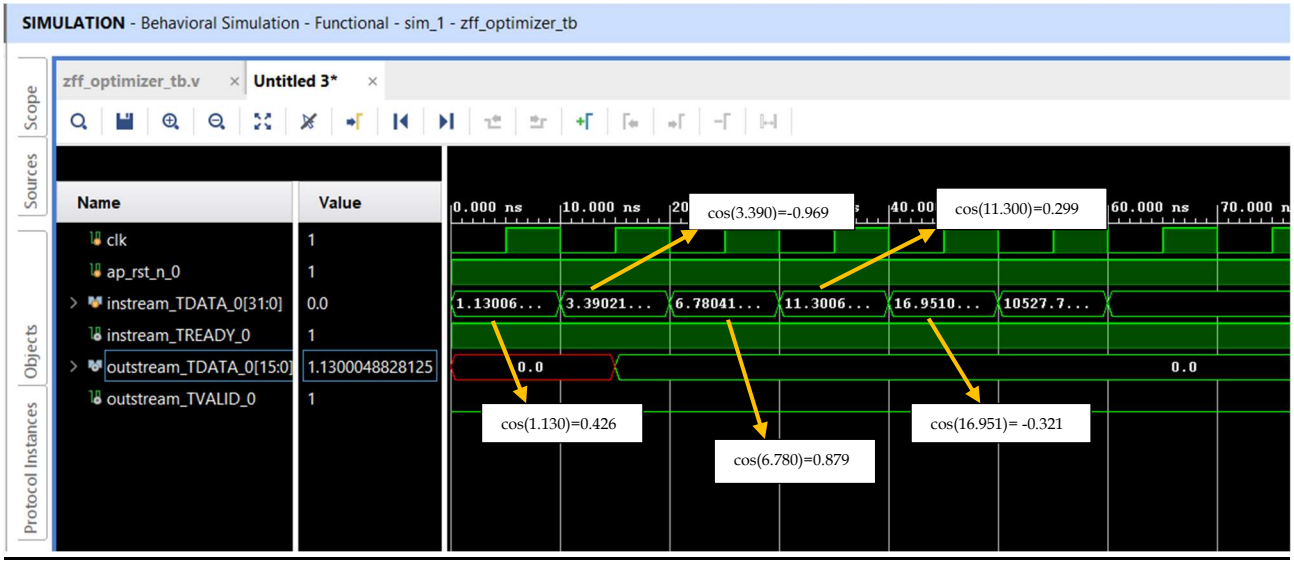
C:/Users/velic/OneDrive/Desktop/vitis-hls/Assignment_9/VIVADO/project_2/project_2.srcs/sim_1/new/zff_optimizer_tb.v

```
1   `timescale 1ns / 1ps
2
3   module zff_optimizer_tb(
4       );
5
6     reg clk;
7     reg ap_rst_n_0;
8     reg [31:0]instream_TDATA_0;
9     wire instream_TREADY_0;
10    wire [15:0]outstream_TDATA_0;
11    wire outstream_TVALID_0;
12
13
14
15    design_2_wrapper
16      optimi_Test(.ap_clk_0(clk),
17       .ap_rst_n_0(ap_rst_n_0),
18       .instream_TDATA_0(instream_TDATA_0),
19       .instream_TREADY_0(instream_TREADY_0),
20       .outstream_TDATA_0(outstream_TDATA_0),
21       .outstream_TVALID_0(outstream_TVALID_0));
22
23
24      always #5 clk=~clk;
25
26  initial
27  begin
28  clk=0;
29  ap_rst_n_0=1'b1;
30  instream_TDATA_0=32'b00000000000000010010000101001100;
31  #10 instream_TDATA_0=32'b00000000000000110110001111100101;
32  #10 instream_TDATA_0=32'b00000000000000110110001111100101;
```

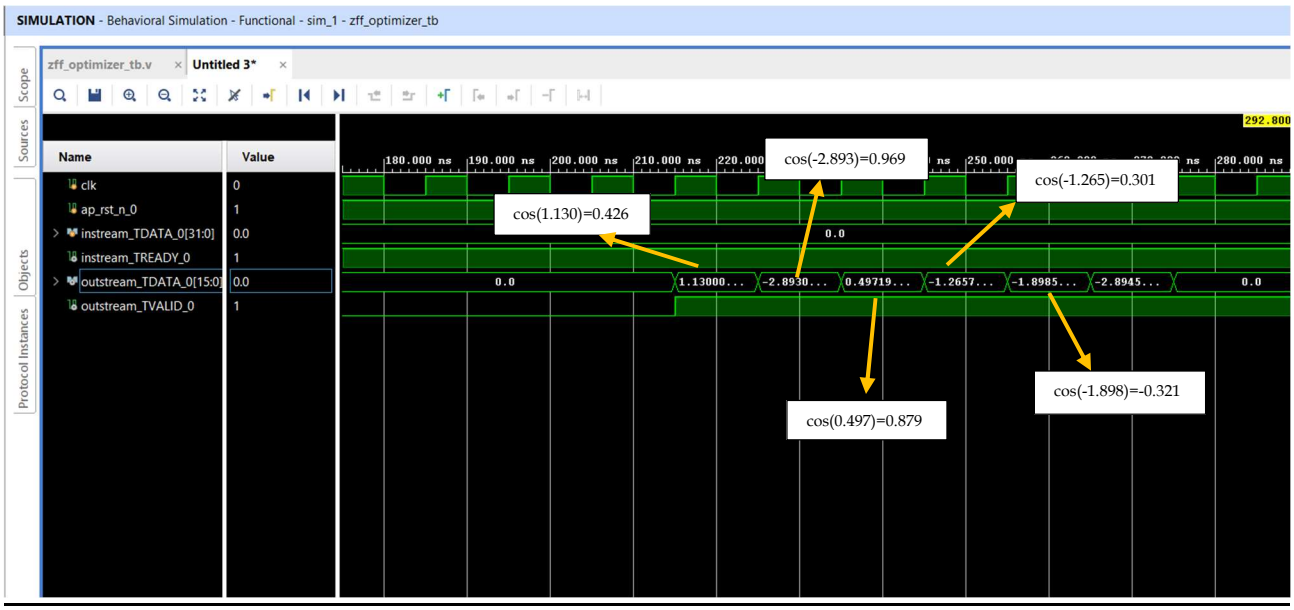
```
33    #10 instream_TDATA_0=32'b00000000000010110100110011111010;
34    #10 instream_TDATA_0=32'b00000000000010000111100110111111;
35    #10  instream_TDATA_0=32'b00101001000111111011100110000000;
36    #10  instream_TDATA_0=32'b0;
37    end
38
39    endmodule
40
```

Simulation Window:

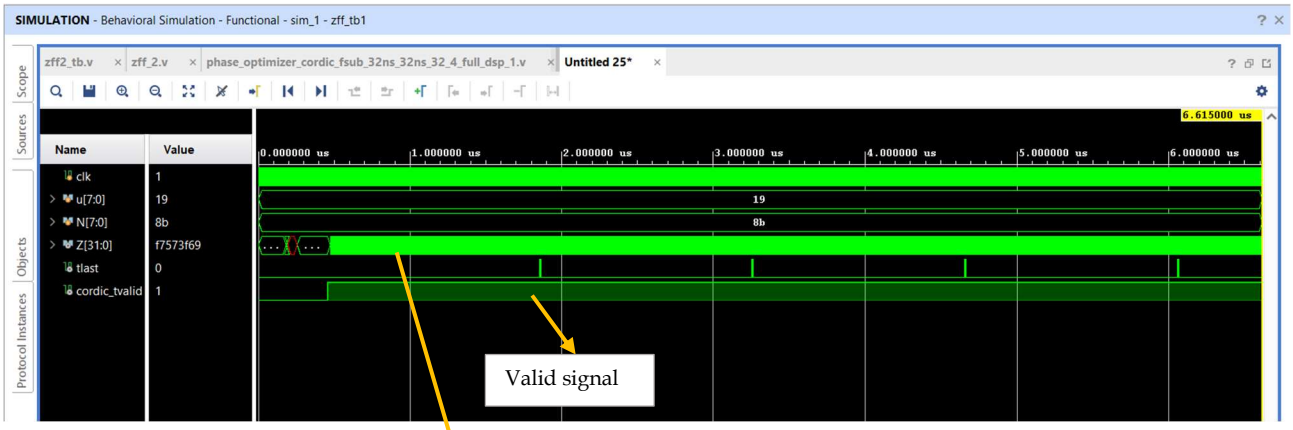
Un-optimized Phase carried from floating point IP's:

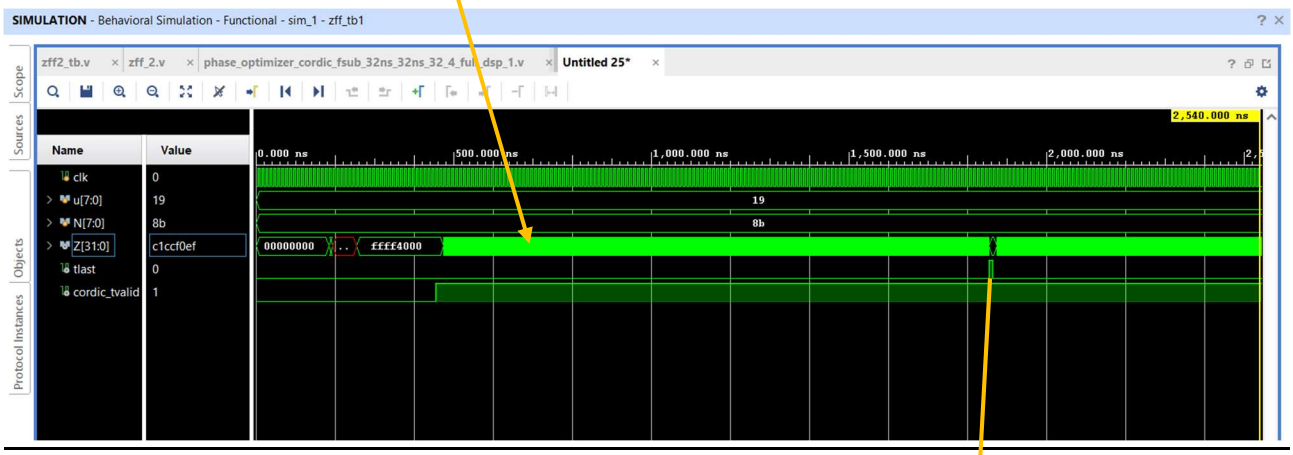


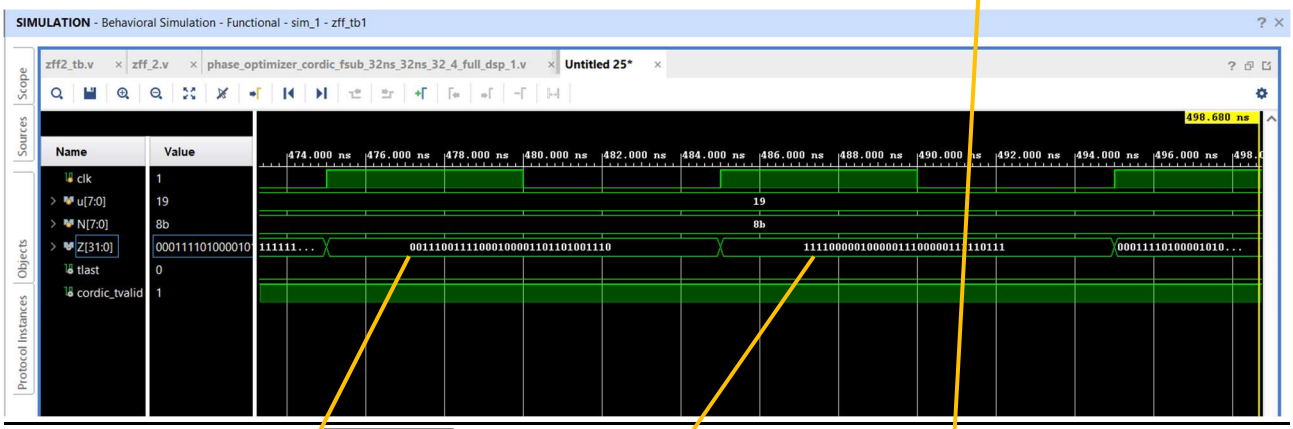
Optimized phase from optimizer:

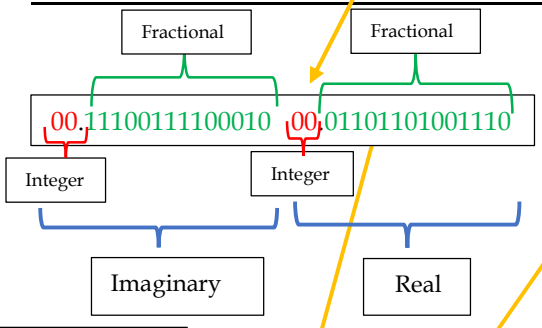


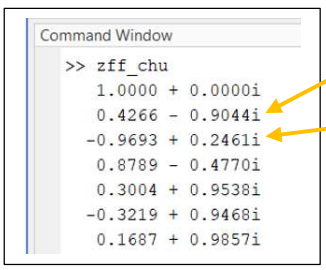
Simulation window: (Vivado Generated IP)





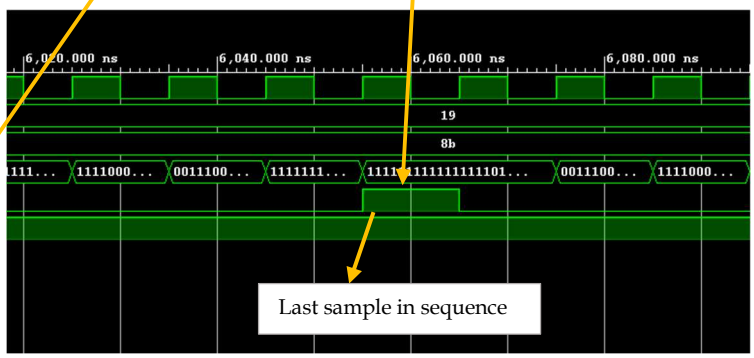






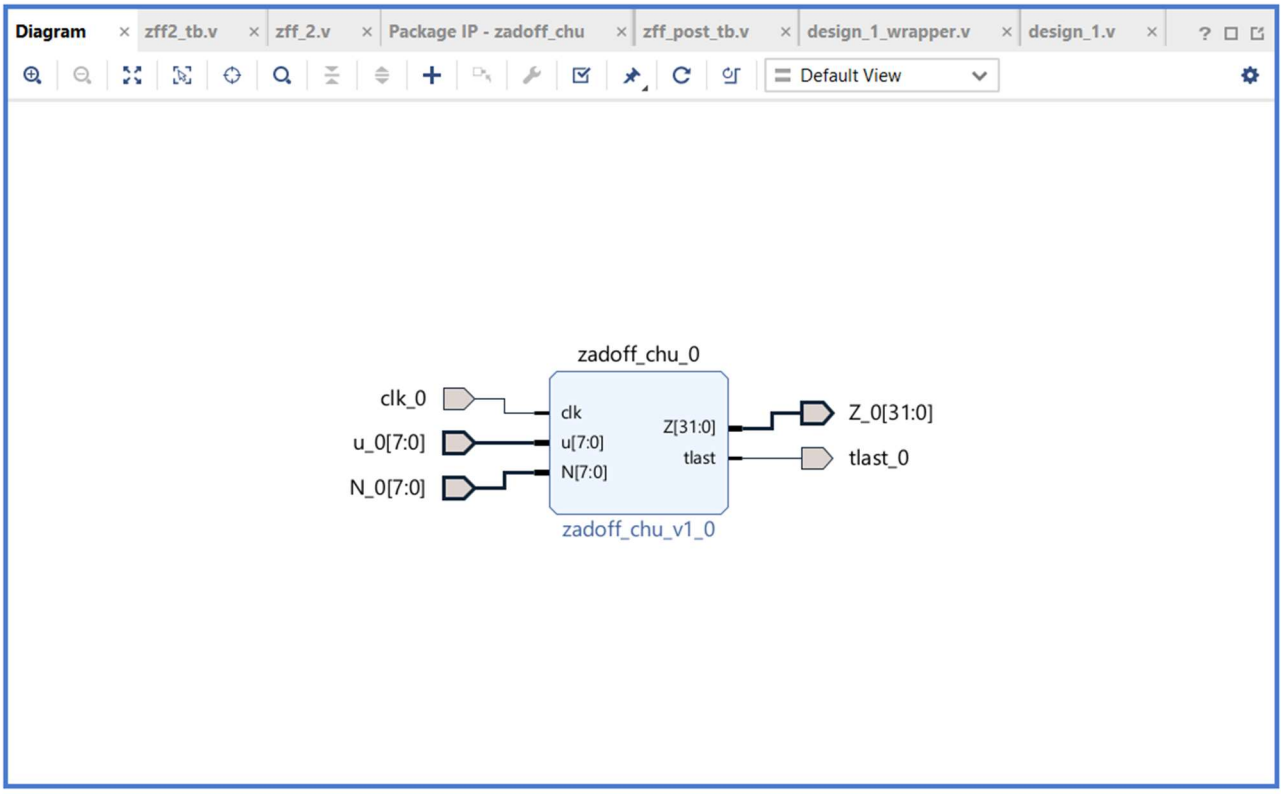
0.4266 - 0.896 *i*

-0.906 + 0.246 *i*

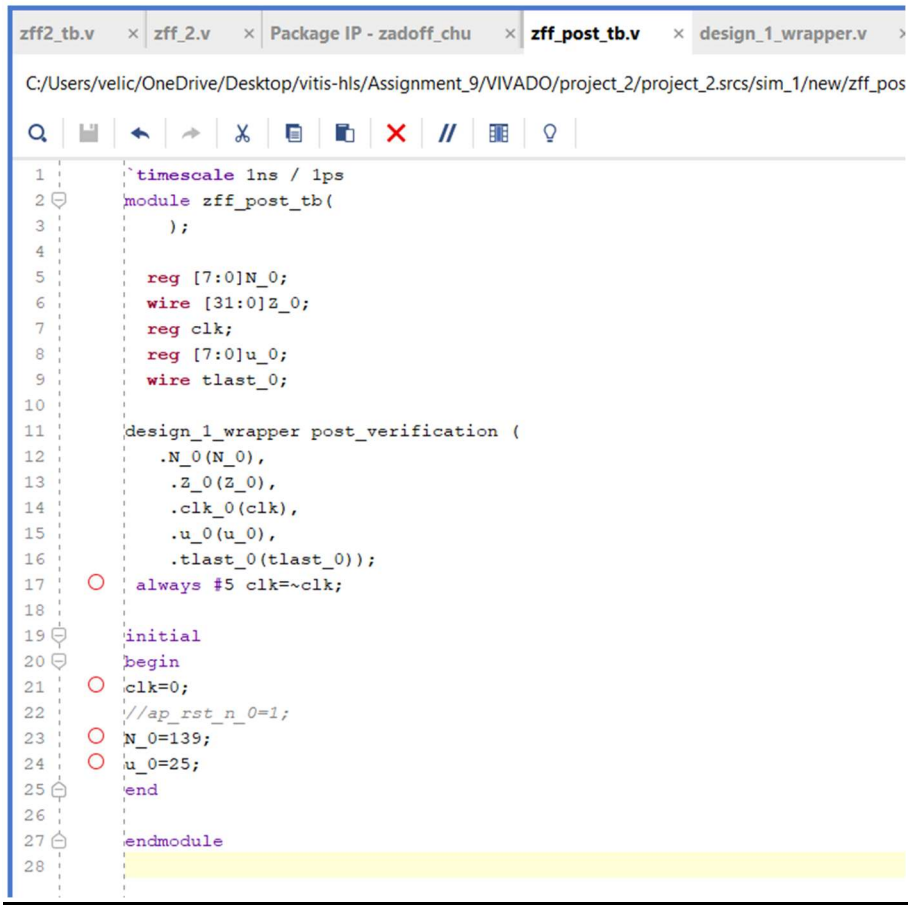


Testing Packaged IP: (Post Implementation direct testing)

Block Design:

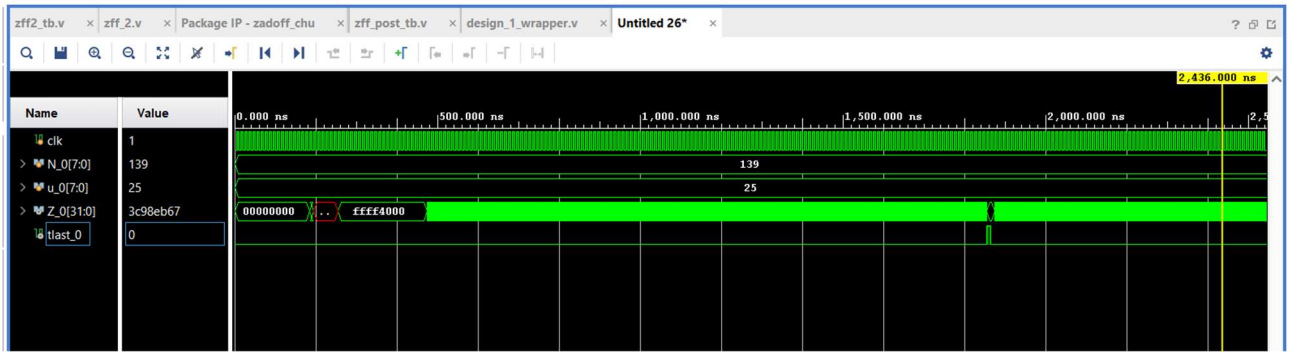


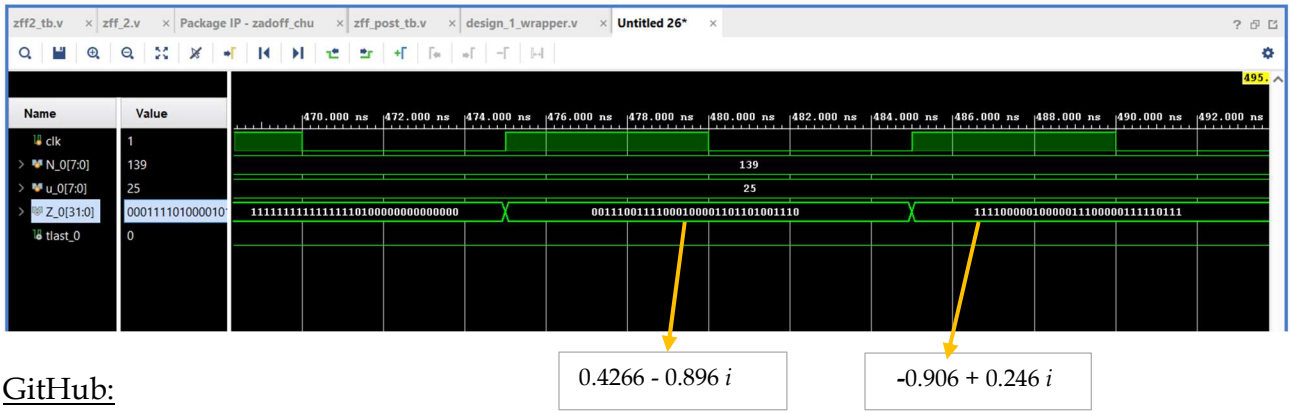
Test Bench:




```
1   `timescale 1ns / 1ps
2   module zff_post_tb(
3       );
4
5     reg [7:0]N_0;
6     wire [31:0]Z_0;
7     reg clk;
8     reg [7:0]u_0;
9     wire tlast_0;
10
11  design_1_wrapper post_verification (
12     .N_0(N_0),
13      .Z_0(Z_0),
14      .clk_0(clk),
15      .u_0(u_0),
16      .tlast_0(tlast_0));
17    always #5 clk=~clk;
18
19  initial
20  begin
21  clk=0;
22  //ap_rst_n_0=1;
23  N_0=139;
24  u_0=25;
25  end
26
27  endmodule
28
```

Simulation Window:





GitHub:

https://github.com/velicharlagokulkumar/vitis-hls/tree/main/Assignment_9

References:

- pg105-cordic
- stack exchange
- https://www.nrexplained.com/sequence
- https://binary-system.base-conversion.ro/convert-real-numbers-from-decimal-system-to-32bit-single-precision-IEEE754-binary-floating-point.php
- https://www.exploringbinary.com/twos-complement-converter/#:~:text=Converting%20Two's%20Complement%20Fixed%2DPoint%20to%20Decimal&text=For%20example%2C%20if%20you%20have,)%2F28%20%3D%20127.99609375.
- https://www.h-schmidt.net/FloatConverter/IEEE754.html
- https://www.vlsiuniverse.com/verilog-code-for-sine-cos-and-tan-cordic/#google_vignette
- https://www.rfwireless-world.com/Terminology/Zadoff-chu-sequence-LTE.html#google_vignette
- https://www.sharetechnote.com/html/Handbook_LTE_Zadoff_Chu_Sequence.html