# Problem 1

## Code lpfirstats()

%lpfirstats(H)

function [ripple, minpass, maxstoplo, maxstophi] = lpfirstats(H)

%H - frequency responce

%100MHz sampling

%passband 8MHz, 12MHz

PB8 = round(length(H) \* 8 / 100);

PB12 = round(length(H) \* 12 / 100);

%stopband 16MHz, 19MHz

SB16 = round(length(H) \* 16 / 100);

SB19 = round(length(H) \* 19 / 100);

%param1 ripple. below 8MHz, < 3dB

ripple = 20\*log10(max(H(1:PB8))) - 20\*log10(min(H(1:PB8))); %log from level 1

%param2 minpass. below 12MH, > -4dB

minpass = 20\*log10(min(H(1:PB12)));

%param3 maxstoplo. above 16MHz, < -25dB

maxstoplo = 20\*log10(max(H(SB16:length(H))));

%param3 maxstophi. above 19MHz, < -28dB

maxstophi = 20\*log10(max(H(SB19:length(H))));

end

## Code for searching parameters smallest area

%search parameters

clear

min\_numtaps = 0;

min\_freqs = [];

min\_amps = [];

min\_scale = 0;

min\_area = -1;

min\_pps = 0;

min\_ripple = 0;

min\_minpass = 0;

min\_stoplo = 0;

min\_stophi = 0;

%passband 8MHz, 12MHz

PB8 = 8 / 100;

PB12 = 12 / 100;

%stopband 16MHz, 19MHz

SB16 = 16 / 100;

SB19 = 19 / 100;

for stopamp = 0:0.05:0.1

for passamp = 0:0.05:0.1

for stopfreq = SB16:0.01:SB19

for passfreq = PB8:0.01:PB12

for scale = 100:100:2048

for numtaps = 31:10:211

freqs = [0 passfreq stopfreq 1];

amps = [1-passamp 1-passamp stopamp stopamp];

coeffs1 = remez(numtaps-1, freqs, amps);

coeffs2 = coeffs1\*scale;

coeffs = round(coeffs2);

[H,W] = freqz(coeffs);

H\_norm = abs(H) ./ abs(H(1));

[ripple, minpass, maxstoplo, maxstophi] = lpfirstats(H\_norm);

if (ripple < 3 && minpass > -4 && maxstoplo < -25 && maxstophi < -28)

fprintf("%d numtaps, %d scale, %f pf, %f sf, %f pa, %f sa CONDITION MET\n", numtaps, scale, passfreq, stopfreq, passamp, stopamp);

%calculate area

n\_pps = 0;

for c = coeffs

n\_pps = n\_pps + numppterms(c);

end

area = n\_pps + 2 \* numtaps;

if area < min\_area || min\_area == -1

min\_area = area;

min\_numtaps = numtaps;

min\_amps = amps;

min\_scale = scale;

min\_freqs = freqs;

min\_pps = n\_pps;

min\_ripple = ripple;

min\_minpass = minpass;

min\_stoplo = maxstoplo;

min\_stophi = maxstophi;

end

else

fprintf("%d numtaps, %d scale, %f pf, %f sf, %f pa, %f sa not met\n", numtaps, scale, passfreq, stopfreq, passamp, stopamp);

end

end

end

end

end

end

end

%report numbers

fprintf("%d area, %d taps, %d PPs\n", min\_area, min\_numtaps, min\_pps);

fprintf("%f ripple, %f minpass, %f maxstoplo, %f maxstophi\n", min\_ripple, min\_minpass, min\_stoplo, min\_stophi);

%3/17/2018 results

%130 area, 41 taps, 50 PPs

%0.616699 ripple, -3.442710 minpass, -25.442571 maxstoplo, -28.018272 maxstophi

%freqs [0 0.09 0.17 1]

%amps [0.9 0.9 0 0]

report\_area = 130;

report\_taps = 41;

report\_pps = 50;

report\_freqs = [0 0.09 0.17 1];

report\_amps = [0.9 0.9 0 0];

## Code numpterms() from hw3

function [mpp] = numppterms(num)

mpp = 0;

rem = num;

while (rem ~= 0 && mpp < 10)

for i = -1:10

if 2^i >= abs(rem)

mpp = mpp + 1;

if (2^i - abs(rem) < abs(rem) - 2^(i-1))

rem = abs(rem) - 2^i;

else

rem = abs(rem) - 2^(i-1);

end

break;

end

end

end

end

## Plot\_one\_lpfir

## Plot stem()

# Problem 2

|  |  |  |  |
| --- | --- | --- | --- |
| Clock period | Area | Clock met | Clock-data difference |
| 1ms |  |  |  |
| 0.1ns |  |  |  |
|  |  |  |  |

## Difff.m result

## Code

## Code for test bench

//tbench for fft

`timescale 1ns/10ps

module tbench();

integer file;

reg [11:0] angle;

reg [15:0] A\_real;

reg [15:0] A\_imag;

reg [15:0] B\_real;

reg [15:0] B\_imag;

wire [15:0] X\_real;

wire [15:0] X\_imag;

wire [15:0] Y\_real;

wire [15:0] Y\_imag;

reg clk, rst, start;

integer count = 0;

reg [31:0] extreme\_A [19:0];

reg [31:0] extreme\_B [19:0];

reg [11:0] extreme\_exp [19:0];

integer timer = 0;

reg signed [15:0] print0, print1;

reg [31:0] A\_buffer [3:0];

reg [31:0] B\_buffer [3:0];

reg [11:0] exp\_buffer [3:0];

fftbtf F1 (

.clk (clk),

.rst (rst),

.start (start),

.wn\_exp (angle),

.A\_R (A\_real),

.A\_I (A\_imag),

.B\_R (B\_real),

.B\_I (B\_imag),

.X\_R (X\_real),

.X\_I (X\_imag),

.Y\_R (Y\_real),

.Y\_I (Y\_imag)

);

initial begin

$recordfile("fft\_test");

$recordvars(tbench);

file = $fopen("complex\_result.m", "w");

clk = 1'b0;

angle = 12'd0;

A\_real = 16'd0;

A\_imag = 16'd0;

B\_real = 16'd0;

B\_imag = 16'd0;

//set extreme cases

extreme\_exp[0] = 12'b0000\_0000\_0000;

extreme\_A[0] = 32'h00000000;

extreme\_B[0] = 32'h00000000;

extreme\_exp[1] = 12'b0100\_0000\_0000;

extreme\_A[1] = 32'h00000000;

extreme\_B[1] = 32'h00000000;

extreme\_exp[2] = 12'b0100\_0000\_0000;

extreme\_A[2] = 32'h00000001;

extreme\_B[2] = 32'h00000001;

extreme\_exp[3] = 12'b0100\_0000\_0000;

extreme\_A[3] = 32'hFFFFFFFF;

extreme\_B[3] = 32'hFFFFFFFF;

extreme\_exp[4] = 12'b0100\_0000\_0000;

extreme\_A[4] = 32'h0000FFFF;

extreme\_B[4] = 32'hFFFF0000;

extreme\_exp[5] = 12'b0000\_0000\_0000;

extreme\_A[5] = 32'hFFFF0000;

extreme\_B[5] = 32'h0000FFFF;

extreme\_exp[6] = 12'b1111\_1111\_1111;

extreme\_A[6] = 32'hFFFFFFFF;

extreme\_B[6] = 32'h00000000;

extreme\_exp[7] = 12'b1000\_0000\_0000;

extreme\_A[7] = 32'h00000000;

extreme\_B[7] = 32'hFFFFFFFF;

extreme\_exp[8] = 12'b0010\_0000\_0000;

extreme\_A[8] = 32'h00010000;

extreme\_B[8] = 32'h00010000;

extreme\_exp[9] = 12'b1110\_0000\_0000;

extreme\_A[9] = 32'h00010002;

extreme\_B[9] = 32'h00010002;

extreme\_exp[10] = 12'b1010\_0000\_0000;

extreme\_A[10] = 32'hFFFF0000;

extreme\_B[10] = 32'hFFFF0000;

extreme\_exp[11] = 12'b1010\_0000\_0000;

extreme\_A[11] = 32'h00007FFF;

extreme\_B[11] = 32'h00007FFF;

extreme\_exp[12] = 12'b0100\_0000\_0000;

extreme\_A[12] = 32'h7FFF7FFF;

extreme\_B[12] = 32'h7FFF7FFF;

extreme\_exp[13] = 12'b1010\_0000\_0000;

extreme\_A[13] = 32'h7FFF7FFF;

extreme\_B[13] = 32'h7FFF7FFF;

extreme\_exp[14] = 12'b0100\_0000\_0000;

extreme\_A[14] = 32'h80008000;

extreme\_B[14] = 32'h80008000;

extreme\_exp[15] = 12'b0000\_0000\_0000;

extreme\_A[15] = 32'h7FFF8000;

extreme\_B[15] = 32'h80007FFF;

extreme\_exp[16] = 12'b1000\_0000\_0000;

extreme\_A[16] = 32'h7FFF8000;

extreme\_B[16] = 32'h80007FFF;

extreme\_exp[17] = 12'b1000\_0000\_0000;

extreme\_A[17] = 32'h80007FFF;

extreme\_B[17] = 32'h7FFF8000;

extreme\_exp[18] = 12'b0000\_0000\_0000;

extreme\_A[18] = 32'h80007FFF;

extreme\_B[18] = 32'h7FFF8000;

extreme\_exp[19] = 12'b1110\_0000\_0000;

extreme\_A[19] = 32'h7FFF7FFF;

extreme\_B[19] = 32'h80008000;

//init buffer

A\_buffer[0] = 31'd0;

A\_buffer[1] = 31'd0;

A\_buffer[2] = 31'd0;

A\_buffer[3] = 31'd0;

B\_buffer[0] = 31'd0;

B\_buffer[1] = 31'd0;

B\_buffer[2] = 31'd0;

B\_buffer[3] = 31'd0;

exp\_buffer[0] = 12'd0;

exp\_buffer[1] = 12'd0;

exp\_buffer[2] = 12'd0;

exp\_buffer[3] = 12'd0;

rst = 1'b1;

#35

rst = 1'b0;

count = 0;

angle = extreme\_exp[0];

{A\_real, A\_imag} = extreme\_A[0];

{B\_real, B\_imag} = extreme\_B[1];

timer = 3;

start = 1'b1;

end

always begin

#10;

clk = ~clk;

end

always @(posedge clk) begin

//$display("done %d\%",count/10);

A\_buffer[3] <= A\_buffer[2];

A\_buffer[2] <= A\_buffer[1];

A\_buffer[1] <= A\_buffer[0];

A\_buffer[0] <= {A\_real, A\_imag};

B\_buffer[3] <= B\_buffer[2];

B\_buffer[2] <= B\_buffer[1];

B\_buffer[1] <= B\_buffer[0];

B\_buffer[0] <= {B\_real, B\_imag};

exp\_buffer[3] <= exp\_buffer[2];

exp\_buffer[2] <= exp\_buffer[1];

exp\_buffer[1] <= exp\_buffer[0];

exp\_buffer[0] <= angle;

start = 1'b0;

if (rst == 0) begin

if (timer > 0) begin

timer = timer - 1;

end else begin

count = count + 1;

if (count < 20) begin

//send next values

{A\_real, A\_imag} = extreme\_A[count];

{B\_real, B\_imag} = extreme\_B[count];

angle = extreme\_exp[count];

end else begin

{A\_real, A\_imag} = $random;

{B\_real, B\_imag} = $random;

angle = $random % 4096;

end

//report

if (count >= 5) begin

$fwrite(file, "wn\_exp(%d+1) = %d; ", count-3, exp\_buffer[3]);

{print0, print1} = A\_buffer[3];

$fwrite(file, "a(%d+1) = %d + 1i \* %d; ", count-3, print0, print1);

{print0, print1} = B\_buffer[3];

$fwrite(file, "b(%d+1) = %d + 1i \* %d; ", count-3, print0, print1);

print0 = X\_real;

print1 = X\_imag;

$fwrite(file, "x(%d+1) = %d + 1i \* %d; ", count-3, print0, print1);

print0 = Y\_real;

print1 = Y\_imag;

$fwrite(file, "y(%d+1) = %d + 1i \* %d;\n", count-3, print0, print1);

timer = 3;

start = 1'b1;

end

if (count > 1000) begin

$finish;

end

end

end

end

endmodule

## Pipelined block diagram

## Cycle period 1ms

### Area report

### Timing report

## Cycle period 0.1ns

### Area report

### Timing report

## Cycle period ns

### Area report

### Timing report