《数字信号处理》测试题

班级： 20通信工程1班 姓名： 孟凡钧 学号： 20434020109

1.读下列程序，填写输出结果。（9分）

clc 输出结果：

x1=0:4;

y1=0:4;

a1=double(x1==2); a1= [0,0,1,0,0]

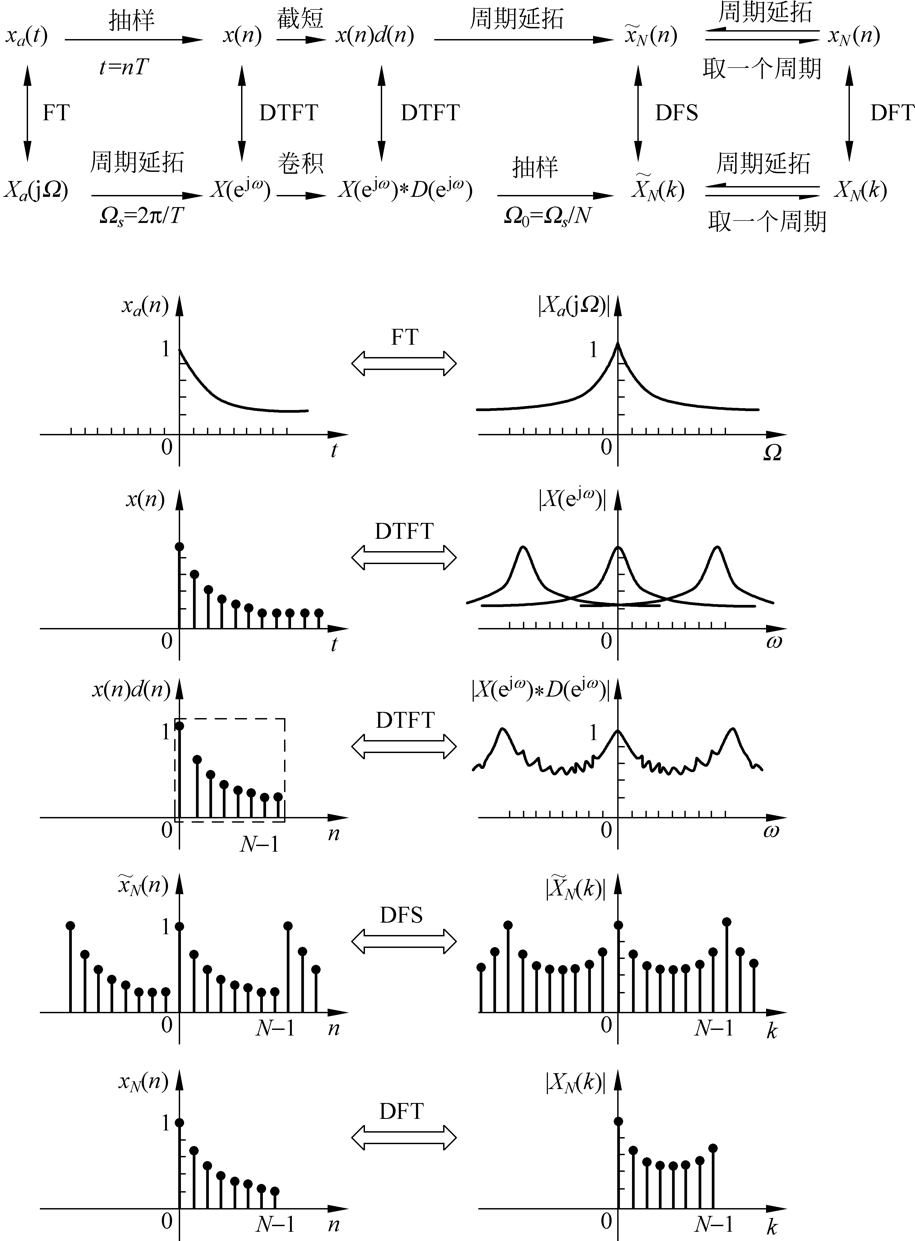
b1=double(y1>2); b1= [0,0,0,1,1]

c1=conv(a1,b1); c1= [0,0,0,0,0,1,1,0,0]

图示

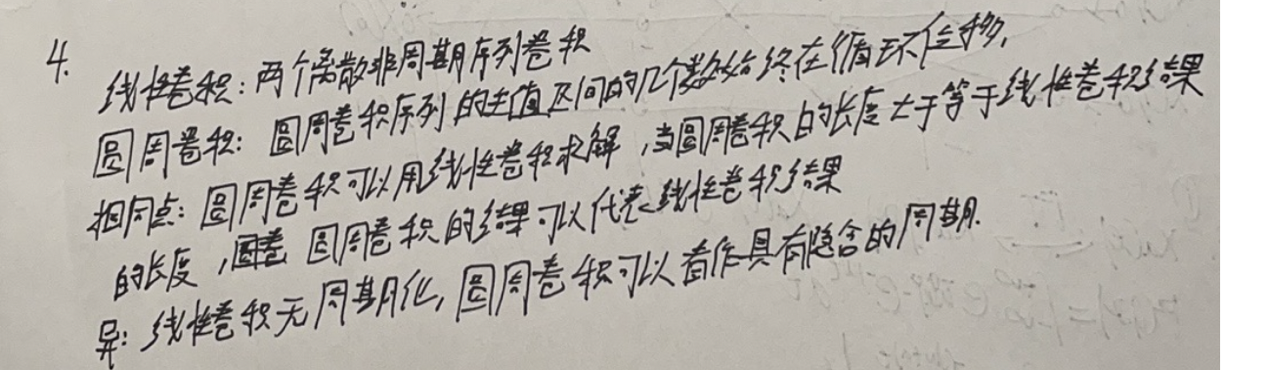
描述已自动生成2.画出*N* = 8点的基-2按时间抽选（DIT）FFT流图（*L* = 3）。（15分）

3. 用DFT对非周期连续时间信号进行频谱分析的过程如下图所示，请根据图中所示时域、频域波形，对每阶段操作的结构进行分析。（25分）

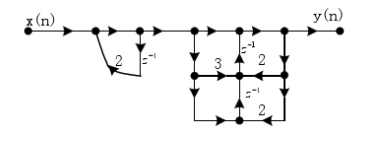


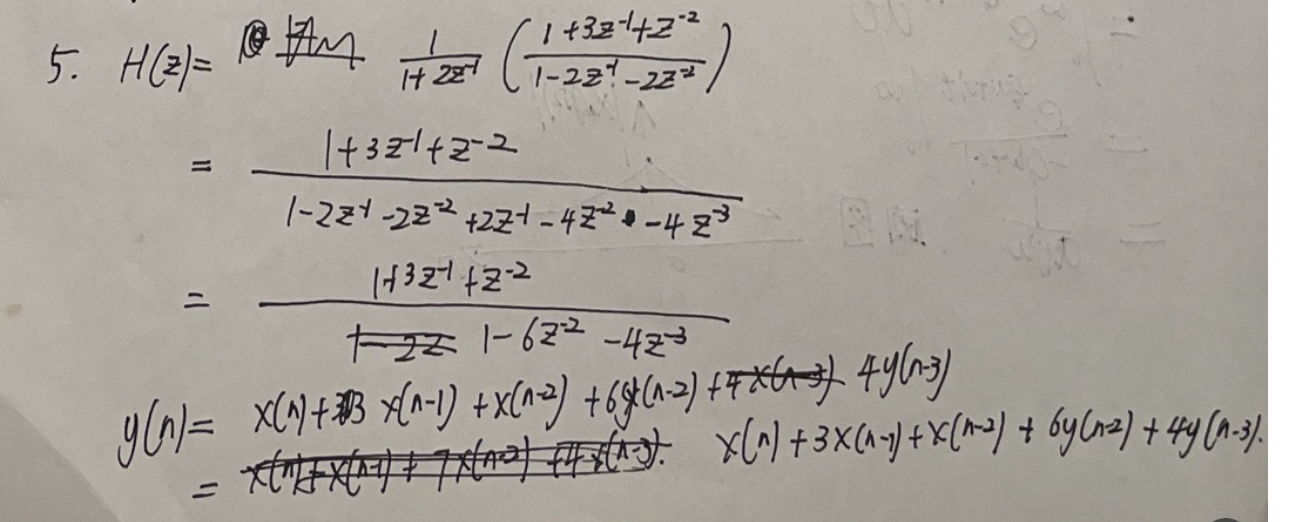
文本, 信件

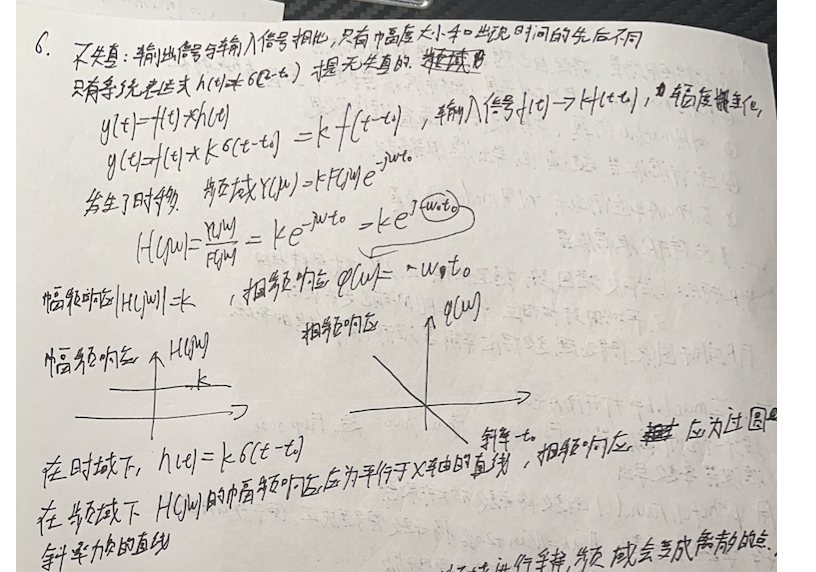
描述已自动生成例：傅里叶变换，其时域为非周期连续信号，经过FT，其频谱为连续函数，对时域信号进行抽样，导致时域信号变为离散时间信号；对时间信号的抽样，造成其频谱\*\*\*\*\*\*\*，此时进行频谱分析使用DTFT；

4. 试说明线性卷积与圆周卷积的异同点。（8分）

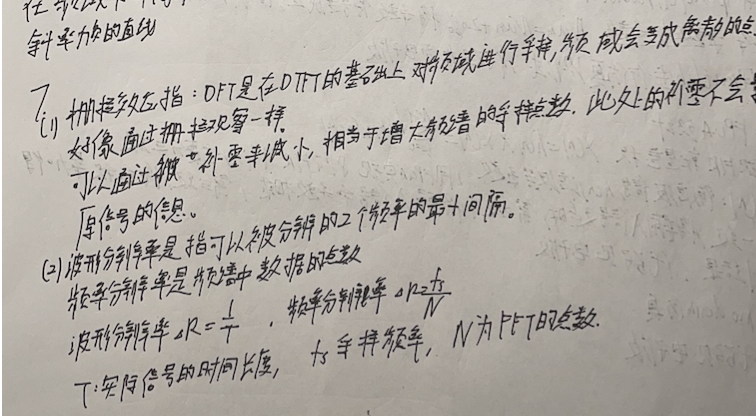
5. 写出下图所示结构的系统函数和差分方程。（8分）





6. 一个不失真线性系统，其幅频响应和相频响应具有什么特点？并运用FT/DFT变换的性质，对其进行解释说明。（10分）

7. （1）请解释栅栏效应，以及改进方法；

（2）请对波形分辨率、频率分辨率的区别进行说明。（15分）

8. 查阅相关文献，总结数字滤波器开发的步骤。（从软件到硬件）（10分）

文本, 信件

描述已自动生成

Matlab仿真测试代码：

>> Num

Num =

-0.032520274817651 -0.038393343465240 0.078426769306951 0.287437638322338 0.398431099816101 0.287437638322338 0.078426769306951 -0.038393343465240 -0.032520274817651

取整：

>> sprintf('%.4f ', Num)

Num=round(Num\*400)%将系数放大并取整

ans =

-0.0325 -0.0384 0.0784 0.2874 0.3984 0.2874 0.0784 -0.0384 -0.0325

Num =

-13 -15 31 115 159 115 31 -15 -13

>> Num=Num+20%将系数符号变成正的，便于FPGA使用

Num =

7 5 51 135 179 135 51 5 7

4，利用matlab进行仿真测试。

clear all；

Fs = 10000; %采样频率决定了两个正弦波点之间的间隔

N = 4096; %采样点数

N1 = 0 : 1/Fs :N/Fs-1/Fs;

x1 = sin(1000\*2\*pi\*N1);

x2 = sin(3000\*2\*pi\*N1);

x3 = sin(4000\*2\*pi\*N1);

in = x1 + x2 + x3;

Fs = 10000; %采样频率决定了两个正弦波点之间的间隔

N = 4096; %采样点数

N1 = 0 : 1/Fs : N/Fs-1/Fs;

s = x1 + x2 + x3;%三种正弦波

fidc = fopen('F:\FPGA\lianxi\FIR\mem.txt','wt'); %将结果写入mem.txt文件，便于modesim使用

for x = 1 : N

fprintf(fidc,'%x\n',round((s(x)+2.12)\*58));

end

fclose(fidc);

a=textread('F:\FPGA\lianxi\FIR\mem.txt','%s');

alpha=hex2dec(a);

coeff =[7,5,51,135,179,135,51,5,7];

out =conv(alpha,coeff);%卷积滤波

subplot(3,1,1);

x4 = round((x1+2.12)\*58);

plot(x4);

axis([0 200 0 256]);

subplot(3,1,2);

plot(alpha);

xlabel('滤波前');

axis([0 200 0 256]);

subplot(3,1,3);

plot(out);

xlabel('滤波后');

axis([0 200 0 160000]);

FPGA滤波器设计代码：

module FIR(

input CLK, //系统时钟50M

input RSTn, //外部复位，Pin31引脚

input [7:0] FIR\_IN,

output reg [17:0] FIR\_OUT

);

reg[7:0] delay\_pipeline1;

reg[7:0] delay\_pipeline2;

reg[7:0] delay\_pipeline3;

reg[7:0] delay\_pipeline4;

reg[7:0] delay\_pipeline5;

reg[7:0] delay\_pipeline6;

reg[7:0] delay\_pipeline7;

reg[7:0] delay\_pipeline8;

reg[7:0] delay\_pipeline9;

always@(posedge CLK or negedge RSTn) begin

if(!RSTn)

begin

delay\_pipeline1 <= 8'b0;

delay\_pipeline2 <= 8'b0;

delay\_pipeline3 <= 8'b0;

delay\_pipeline4 <= 8'b0;

delay\_pipeline5 <= 8'b0;

delay\_pipeline6 <= 8'b0;

delay\_pipeline7 <= 8'b0;

delay\_pipeline8 <= 8'b0;

delay\_pipeline9 <= 8'b0;

end

else

begin

delay\_pipeline1 <= FIR\_IN;

delay\_pipeline2 <= delay\_pipeline1;

delay\_pipeline3 <= delay\_pipeline2;

delay\_pipeline4 <= delay\_pipeline3;

delay\_pipeline5 <= delay\_pipeline4;

delay\_pipeline6 <= delay\_pipeline5;

delay\_pipeline7 <= delay\_pipeline6;

delay\_pipeline8 <= delay\_pipeline7;

delay\_pipeline9 <= delay\_pipeline8;

end

end

wire[7:0] coeff1 = 8'd7; //滤波器系数

wire[7:0] coeff2 = 8'd5;

wire[7:0] coeff3 = 8'd51;

wire[7:0] coeff4 = 8'd135;

wire[7:0] coeff5 = 8'd179;

wire[7:0] coeff6 = 8'd135;

wire[7:0] coeff7 = 8'd51;

wire[7:0] coeff8 = 8'd5;

wire[7:0] coeff9 = 8'd7;

reg signed [16:0] multi\_data1;//乘积结果

reg signed [16:0] multi\_data2;

reg signed [16:0] multi\_data3;

reg signed [16:0] multi\_data4;

reg signed [16:0] multi\_data5;

reg signed [16:0] multi\_data6;

reg signed [16:0] multi\_data7;

reg signed [16:0] multi\_data8;

reg signed [16:0] multi\_data9;

always@(posedge CLK or negedge RSTn) begin//x(1) \* h(1)

if(!RSTn)

multi\_data1 <= 17'b0;

else

multi\_data1 <= delay\_pipeline1\*coeff1;

end

always@(posedge CLK or negedge RSTn) begin//x(1) \* h(1)

if(!RSTn)

multi\_data2 <= 17'b0;

else

multi\_data2 <= delay\_pipeline2\*coeff2;

end

always@(posedge CLK or negedge RSTn) begin//x(1) \* h(1)

if(!RSTn)

multi\_data3 <= 17'b0;

else

multi\_data3 <= delay\_pipeline3\*coeff3;

end

always@(posedge CLK or negedge RSTn) begin//x(1) \* h(1)

if(!RSTn)

multi\_data4 <= 17'b0;

else

multi\_data4 <= delay\_pipeline4\*coeff4;

end

always@(posedge CLK or negedge RSTn) begin//x(1) \* h(1)

if(!RSTn)

multi\_data5 <= 17'b0;

else

multi\_data5 <= delay\_pipeline5\*coeff5;

end

always@(posedge CLK or negedge RSTn) begin//x(1) \* h(1)

if(!RSTn)

multi\_data6 <= 17'b0;

else

multi\_data6 <= delay\_pipeline6\*coeff6;

end

always@(posedge CLK or negedge RSTn) begin//x(1) \* h(1)

if(!RSTn)

multi\_data7 <= 17'b0;

else

multi\_data7 <= delay\_pipeline7\*coeff7;

end

always@(posedge CLK or negedge RSTn) begin//x(1) \* h(1)

if(!RSTn)

multi\_data8 <= 17'b0;

else

multi\_data8 <= delay\_pipeline8\*coeff8;

end

always@(posedge CLK or negedge RSTn) begin//x(1) \* h(1)

if(!RSTn)

multi\_data9 <= 17'b0;

else

multi\_data9 <= delay\_pipeline9\*coeff9;

end

//===================================================================

//加法器

//====================================================================

always@(posedge CLK or negedge RSTn) begin

if(!RSTn)

FIR\_OUT <= 18'b0;

else

FIR\_OUT <= multi\_data1 + multi\_data2 + multi\_data3 + multi\_data4 +multi\_data5 + multi\_data6 + multi\_data7 + multi\_data8 + multi\_data9;

end

endmodule

modelsim代码：

module FIR\_tb();

reg CLK;

reg [7:0] FIR\_IN;

reg RSTn;

reg [7:0] mem[1:4096];

wire [17:0] FIR\_OUT;

reg [12:0] i;

flow\_led u0\_flow\_led (

.CLK(CLK),

.FIR\_IN(FIR\_IN),

.FIR\_OUT(FIR\_OUT),

.RSTn(RSTn)

);

initial

begin

$readmemh("F:/FPGA/lianxi/FIR/mem.txt",mem);//????????mem

RSTn= 0;

CLK= 0;

#50;

RSTn= 1;

#50000;

$stop;

end

initial

forever

#10 CLK = ~CLK;

always@(posedge CLK or negedge RSTn)

if(!RSTn)

FIR\_IN <= 8'b0 ;

else

FIR\_IN <= mem[i];

always@(posedge CLK or negedge RSTn)

if(!RSTn)

i <= 12'd0;

else

i <= i + 1'd1;

endmodule