



茶月猫的博客

http://blog.sina.com.cn/u/5289241396 [订阅] [手机订阅]

首页 博文目录 图片 关于我

个人资料

正文

字体大小: 大 中 小



茶月猫

微博

加好友 发纸条
写留言 加关注



博客等级: 8
博客积分: 21
博客访问: 2,012
关注人气: 1
获赠金笔: 0
赠出金笔: 1
荣誉徽章:

相关博文

- 笑死人不偿命的搞笑动物（组图）
沙漠胡杨树sd08
- 海之南：登上分界洲岛（图）
天山海草
- 但，那已不是我
蛇影
- 「粘土人莱昂纳多·沃奇」◎
goodsmile上海
- 2283清理廢氣與雜質
海雲繼夢
- 《俄罗斯族巴扬艺术》非物质文化遗产
伊犁老狗
- 看淡世事沧桑，内心安然无恙
沙漠胡杨树sd08
- 农村人为啥看北京人傻敢要北京人
关山枫叶

基于Verilog语言的FIR滤波【程序和理解】

(2016-07-08 11:10:41)

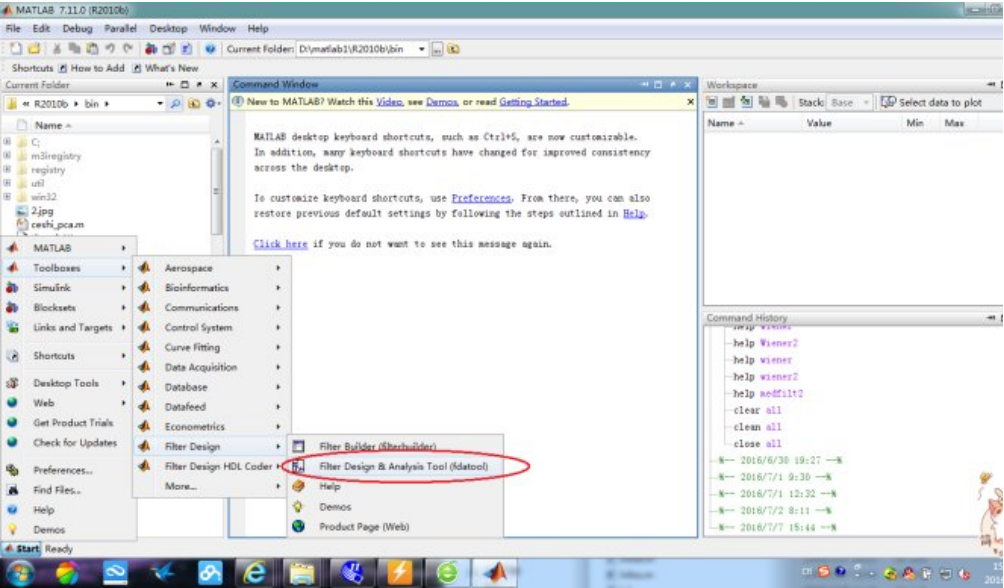
转载 ▼

标签: fpga it 杂谈 分类: FPGA

一直想找一个简单、清晰、明了的fir滤波器的设计，终于找到了一个可以应用的，和大家分享一下，有助于FPGA新手入门。

1.说道fir滤波器，滤波系数肯定是最重要的，因为后面程序中涉及到滤波系数问题，所以先来介绍，此处使用matlab来辅助求出。

①打开matlab中的start，toolbox，filter design，filter design & Analysis Tool，具体位置见下图。



②选择想要涉及的滤波器类型，本次以8阶fir滤波器为例。

设计参数：低通fir滤波器，采样精度是根据自己的输入数据来的，本例为25MHz，通过频率2MHz，截止频率8MHz，可以在specify order处选择几阶滤波器。

印度之旅——印度美女名不虚传
毛毛雨

春之絮语（一）
夜色霓虹

更多>>

推荐博文

想去雄安新区炒房？还是歇歇吧

安倍为何宁赔夫人也不“折兵”

从裸贷到卖卵，到底还有多少坑等

北上广互怼

高娓娓：深层次地八一八特朗普一

破解执行难先要拿“官赖”开刀

密会、窃听、黑金：“通俄门”困

浮夸风从温州小镇吹起来了？

坟头的失守和房事的焦虑为何愈演

你不是徐静蕾，可别被新女权给毁



花婆节祈福接福吃粥福



气质忧郁迷幻的美女



走在世界最美的公路



玉渊潭含苞待放的晚樱



可爱的小松鼠

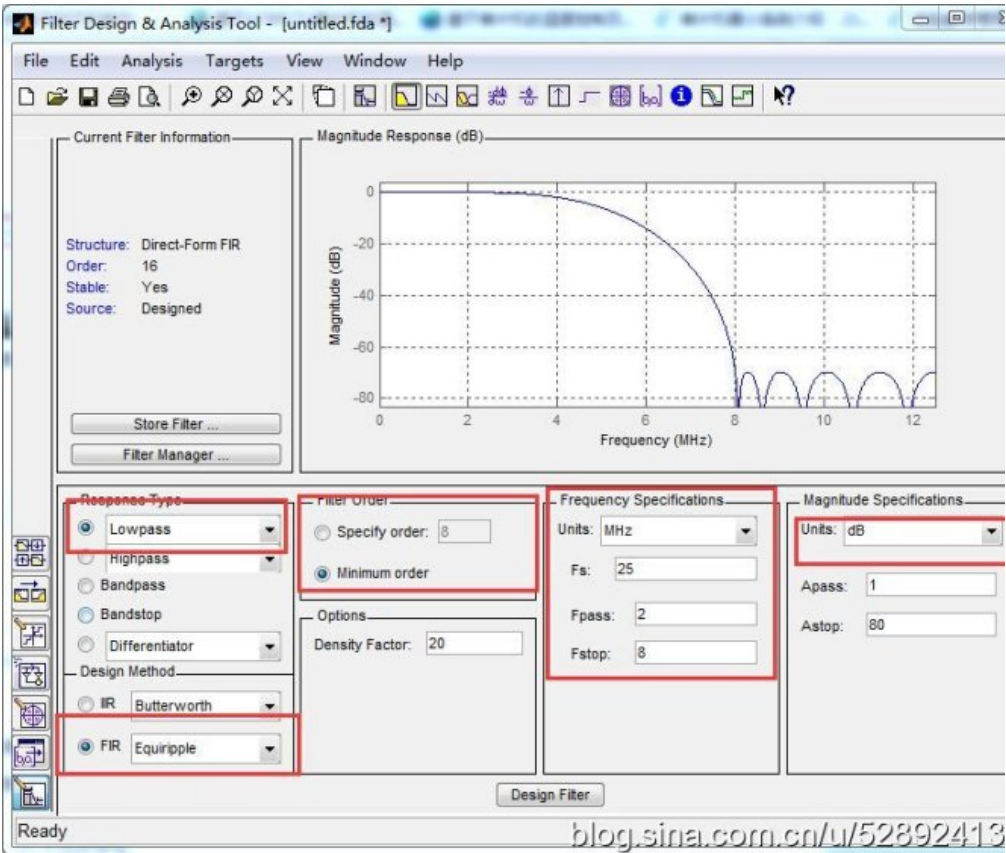


神驹4小时跑完80公里

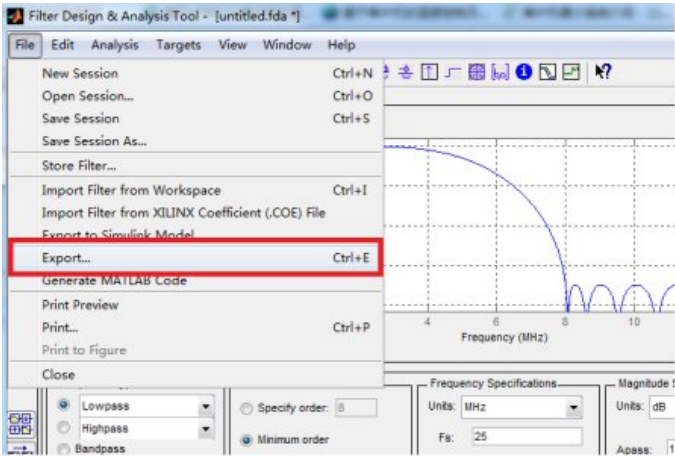
查看更多>>

谁看过这篇博文

njjnwsy	4月10日
海蓝天清	4月8日
浮木无文	4月4日
觥为酣	4月3日
Z_syphus	3月30日
用户38167...	3月28日
egg126	3月26日
沃夫道格	3月22日
SeraphJacky	3月17日
smile	3月13日
嗜睡玄帝	3月6日
牛醉醉	3月1日



③把滤波器数据导出，选择export，在随后弹出的框中再次点击Export（本步骤可以改变数据的变量名），就可以看到命名为Num的滤波器系数出现在目录里。



Name	Value	Min	Max
COF	<17x1 double>	-1435	13133
COEFFICIENT	<17x1 double>	0.04...	0.4008
Num	[0.0073 0.0460 0.13...	0.0073	0.2897
a1	<17x1 double>	-0.04...	0.4008
width	16	16	16

④在matlab框中输入指令Num=Num'

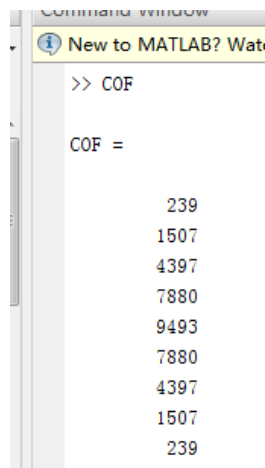
可以看到如下结果，这个就是滤波器的系数了，新建一个txt文件，命名如图，把滤波器系数复制进去。



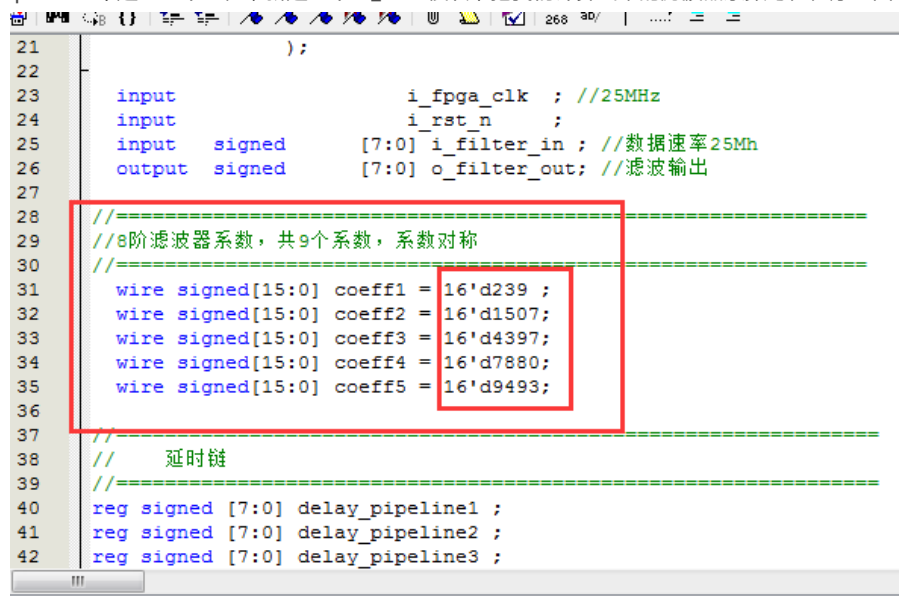
⑤编写matlab程序，进行系数量化。

运行下面一段程序，生成的COF就是最后的量化数据，记录这组数据。

```
clc;
clear all;
load COFFICIENT.dat;%加载系数
a1=COFFICIENT(1:1:length(COFFICIENT));
width = 16;%数据宽度8位
% 量化滤波器系数
COF = round(a1 .* (2^(width-1) - 1));%量化正弦波形数据并取整
```



2.在quartusII中建立一个工程，新建一个fir_filter模块，把我们计算出来的滤波器系数写在程序里面



整体程序如下：

```
`timescale 1 ns / 1 ns
module fir_filter
```

```

(
    i_fpga_clk ,
    i_rst_n ,
    i_filter_in,
    o_filter_out
);

input          i_fpga_clk ; //25MHz
input          i_rst_n  ;
input  signed   [7:0] i_filter_in ; //数据速率25Mh
output signed   [7:0] o_filter_out; //滤波输出

//=====
//8阶滤波器系数，共9个系数，系数对称
//=====
wire signed[15:0] coeff1 = 16'd239 ;
wire signed[15:0] coeff2 = 16'd1507;
wire signed[15:0] coeff3 = 16'd4397;
wire signed[15:0] coeff4 = 16'd7880;
wire signed[15:0] coeff5 = 16'd9493;

//=====
// 延时链
//=====
reg signed [7:0] delay_pipeline1 ;
reg signed [7:0] delay_pipeline2 ;
reg signed [7:0] delay_pipeline3 ;
reg signed [7:0] delay_pipeline4 ;
reg signed [7:0] delay_pipeline5 ;
reg signed [7:0] delay_pipeline6 ;
reg signed [7:0] delay_pipeline7 ;
reg signed [7:0] delay_pipeline8 ;

always@(posedge i_fpga_clk or negedge i_rst_n)
if(!i_rst_n)
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 ;
end
else
begin
    delay_pipeline1 <= i_filter_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 ;
end

//=====
//加法，对称结构，减少乘法器的数目
//=====
reg signed [8:0] add_data1 ;
reg signed [8:0] add_data2 ;
reg signed [8:0] add_data3 ;
reg signed [8:0] add_data4 ;
reg signed [8:0] add_data5 ;

always@(posedge i_fpga_clk or negedge i_rst_n) //x(0)+x(8)
if(!i_rst_n)
    add_data1 <= 9'b0 ;

```

```

else
    add_data1 <= i_filter_in + delay_pipeline8 ;
always@(posedge i_fpga_clk or negedge i_rst_n) //x(1)+x(7)
if(!i_rst_n)
    add_data2 <= 9'b0 ;
else
    add_data2 <= delay_pipeline1 + delay_pipeline7 ;
always@(posedge i_fpga_clk or negedge i_rst_n) //x(2)+x(6)
if(!i_rst_n)
    add_data3 <= 9'b0 ;
else
    add_data3 <= delay_pipeline2 + delay_pipeline6 ;

always@(posedge i_fpga_clk or negedge i_rst_n) //x(3)+x(5)
if(!i_rst_n)
    add_data4 <= 9'b0 ;
else
    add_data4 <= delay_pipeline3 + delay_pipeline5 ;
always@(posedge i_fpga_clk or negedge i_rst_n) //x(4)
if(!i_rst_n)
    add_data5 <= 9'b0 ;
else
    add_data5 <= {delay_pipeline4[7],delay_pipeline4} ;

//=====
//乘法器
//=====
reg signed [24:0] multi_data1 ;
reg signed [24:0] multi_data2 ;
reg signed [24:0] multi_data3 ;
reg signed [24:0] multi_data4 ;
reg signed [24:0] multi_data5 ;

always@(posedge i_fpga_clk or negedge i_rst_n) // ( x(0)+x(8) ) *h(0)
if(!i_rst_n)
    multi_data1 <= 24'b0 ;
else
    multi_data1 <= add_data1*coeff1 ;
always@(posedge i_fpga_clk or negedge i_rst_n) // ( x(1)+x(7) ) *h(1)
if(!i_rst_n)
    multi_data2 <= 24'b0 ;
else
    multi_data2 <= add_data2*coeff2 ;
always@(posedge i_fpga_clk or negedge i_rst_n) // ( x(2)+x(6) ) *h(2)
if(!i_rst_n)
    multi_data3 <= 24'b0 ;
else
    multi_data3 <= add_data3*coeff3 ;
always@(posedge i_fpga_clk or negedge i_rst_n) // ( x(0)+x(8) ) *h(3)
if(!i_rst_n)
    multi_data4 <= 24'b0 ;
else
    multi_data4 <= add_data4*coeff4 ;
always@(posedge i_fpga_clk or negedge i_rst_n) //x(4)*h(4)
if(!i_rst_n)
    multi_data5 <= 24'b0 ;
else
    multi_data5 <= add_data5*coeff5 ;

//=====
//流水线累加
//=====
reg signed[25:0] add_level1_1;//1级
reg signed[25:0] add_level1_2;//1级

```

```

reg signed[25:0] add_level1_3;//1级
always@(posedge i_fpga_clk or negedge i_rst_n) // ( x(0)+x(8) ) *h(0)+ ( x(1)+x(7) ) *h(1)
if(!i_rst_n)
    add_level1_1 <= 26'b0 ;
else
    add_level1_1 <= multi_data1+multi_data2 ;
always@(posedge i_fpga_clk or negedge i_rst_n) // ( x(2)+x(6) ) *h(2)+ ( x(3)+x(5) ) *h(3)
if(!i_rst_n)
    add_level1_2 <= 26'b0 ;
else
    add_level1_2 <= multi_data3+multi_data4 ;
always@(posedge i_fpga_clk or negedge i_rst_n) //x(4)*h(4)
if(!i_rst_n)
    add_level1_3 <= 26'b0 ;
else
    add_level1_3 <= {multi_data5[24],multi_data5} ;

//==2级加法
reg signed [26:0] add_level2_1 ;
reg signed [26:0] add_level2_2 ;
always@(posedge i_fpga_clk or negedge i_rst_n) // ( x(0)+x(8) ) *h(0)+ ( x(1)+x(7) ) *h(1)+ ( x(2)+x(6) )
*h(2)+ ( x(3)+x(5) ) *h(3)
if(!i_rst_n)
    add_level2_1 <= 27'b0 ;
else
    add_level2_1 <= add_level1_1+add_level1_2 ;
always@(posedge i_fpga_clk or negedge i_rst_n) //x(4)*h(4)
if(!i_rst_n)
    add_level2_2 <= 27'b0 ;
else
    add_level2_2 <= {add_level1_3[25],add_level1_3} ;

//-==3级
reg signed [27:0] add_level3_1 ;
always@(posedge i_fpga_clk or negedge i_rst_n)
if(!i_rst_n)
    add_level3_1 <= 27'b0 ;
else
    add_level3_1 <= add_level2_1+add_level2_2 ;

//=====
// 5、output
//=====
reg signed [22:0] r_filter_out ;

if(!i_rst_n)
    r_filter_out <= 23'b0 ;
else
    r_filter_out <= (add_level3_1[22:0]+!add_level3_1[22],{14{add_level3_1[22]}})>>15 ;//四舍五入输出

//=====
// 6、output 取低8位
//=====
assign o_filter_out = r_filter_out[7:0] ;

endmodule

```

推荐：关于枸杞的若干条真相 细节里看出爱不爱你 [×](#)

[新浪首页](#) [登录](#) [注册](#)

*****总结*****

因为输入的数据是AD芯片采样的结果，该AD的采样精度是8位，所以本例使用8阶滤波器，设计的延时链、加法、乘法的程序都是根据8位来的。所以，如果数据输入是16位，或32位等等，需要改变的有设计滤波器的系数等等。

相应的延时链要多添加至reg signed [7:0] delay_pipeline16；

加法和乘法也要有相应的改变，举个例子，大家自行修改

```

always@(posedge i_fpga_clk or negedge i_rst_n) //x(0)+x(16)
if(!i_rst_n)

```

```
add_data1 <= 9'b0 ;
else
add_data1 <= i_filter_in + delay_pipeline16 ;
```

50

喜欢赠金笔

分享：

阅读(542) | 评论 (0) | 收藏(0) | 转载(1) | 喜欢▼ | 打印 | 举报

已投稿到： 排行榜

前一篇：quartusII 怎么配置IP核？【以FFT为例】

后一篇：Verilog实现pwm波控制灯的亮暗【FPGA】

评论

重要提示：警惕虚假中奖信息

[发评论]

做第一个评论者吧！

抢沙发>>

发评论

[更多>>](#)





登录名： 密码： [找回密码](#) [注册](#) ☒ 记住登录状态

☒ 分享到微博 ☐ 评论并转载此博文

按住左边滑块，拖动完成上方拼图

发评论

以上网友发言只代表其个人观点，不代表新浪网的观点或立场。

< 前一篇

quartusII 怎么配置IP核？【以FFT为例】

后一篇 >

Verilog实现pwm波控制灯的亮暗【FPGA】

