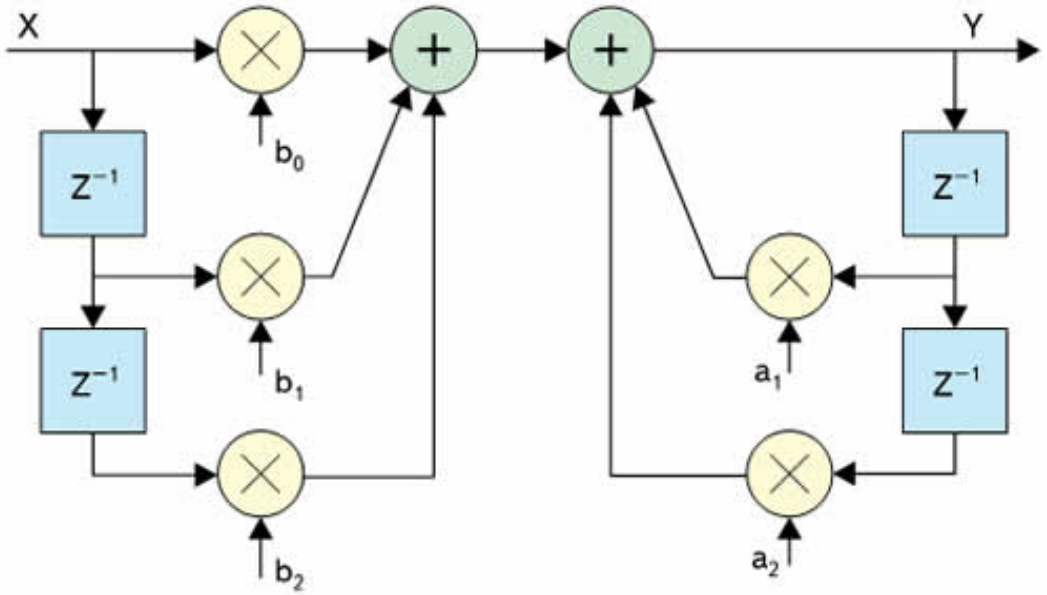
Математическое моделирование РТУ и С

Лекция 8. Проектирование цифровых фильтров



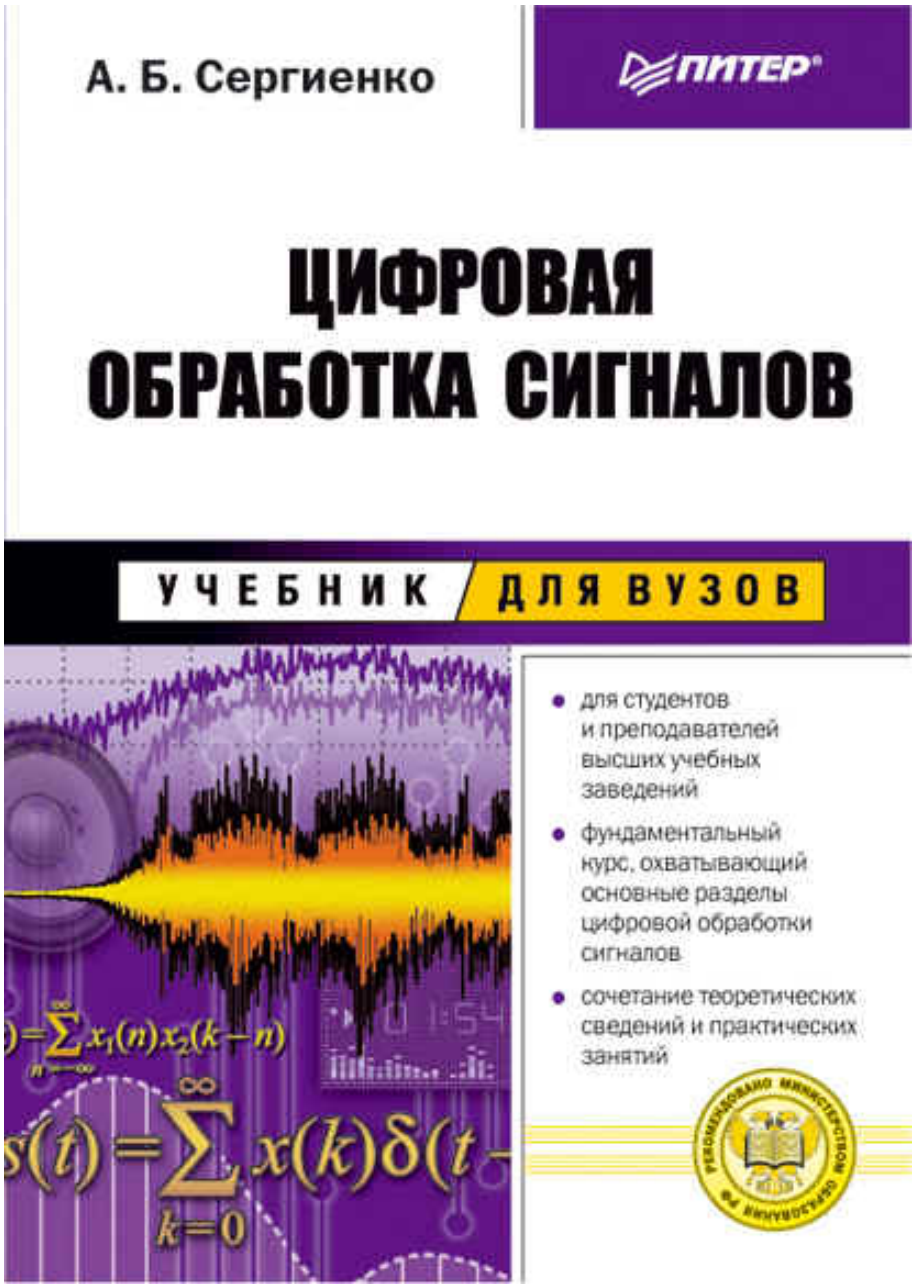
Преподаватель:
Корогодин Илья
korogodin@srns.ru

Литература

А.Б.Сергиенко. Цифровая обработка сигналов. СПб, Питер, 2002. — 608 с.: ил.

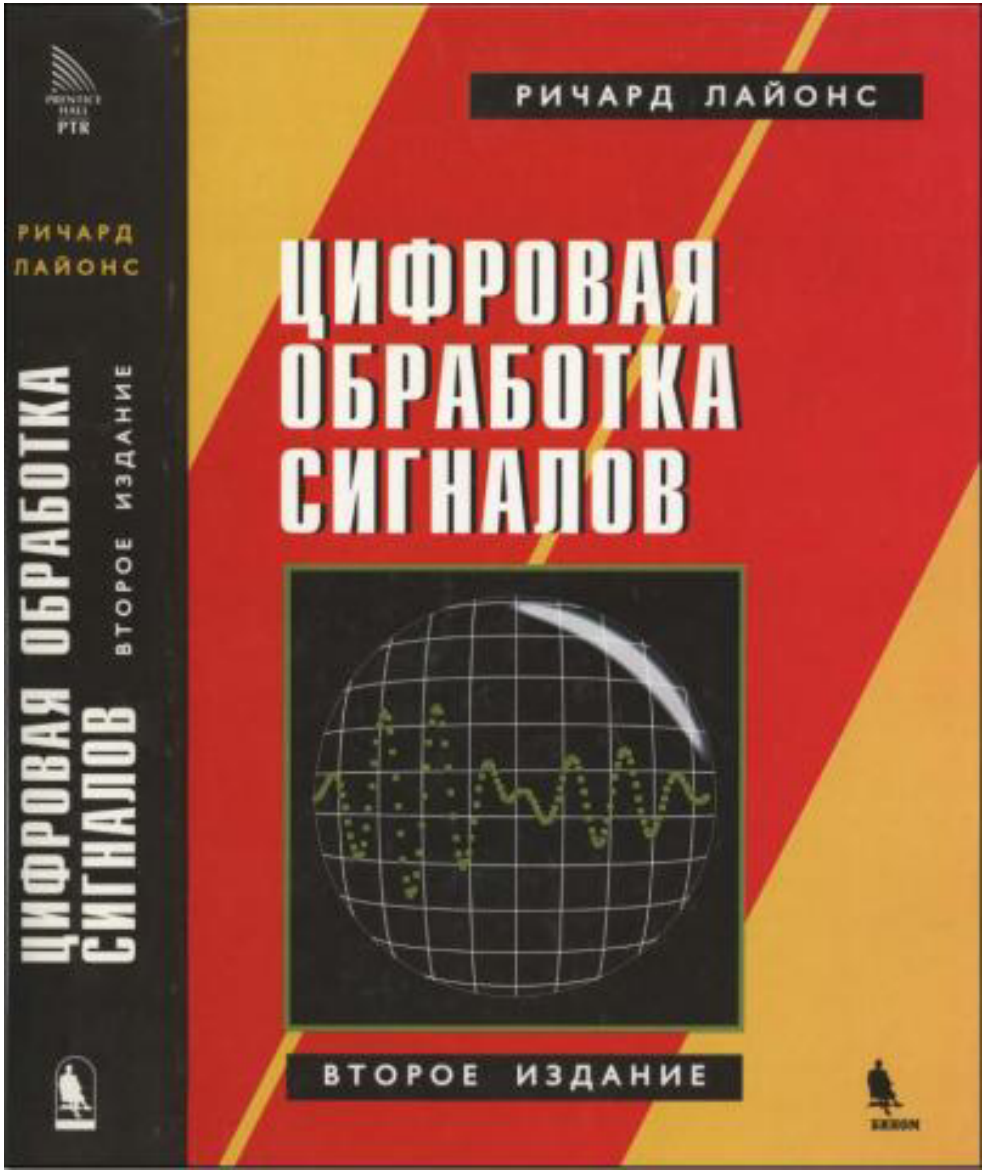
Глава 2. Аналоговые системы

Глава 6. Проектирование дискретных фильтров



Литература

Ричард Лайонс - Цифровая обработка сигналов / Understanding Digital Signal Processing, 2006



Глава 5. Фильтры с импульсной характеристикой конечной длины

Глава 6. Фильтры с импульсной характеристикой бесконечной длины

Глава 7. Специальные КИХ-фильтры нижних частот

Задача проектирования

Возникают задачи:

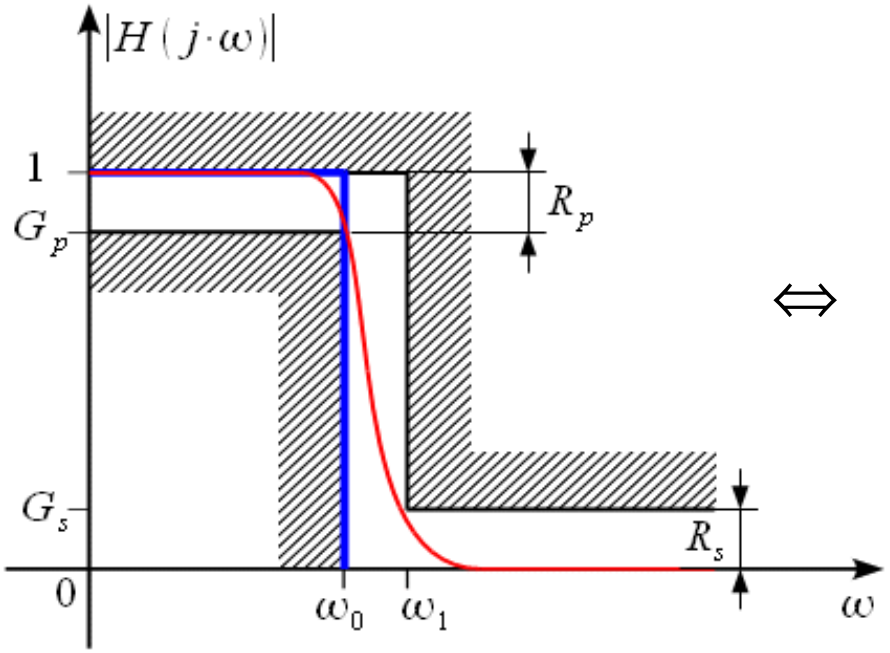
- собрать модель по основным характеристикам АЧХ/ФЧХ;

- выяснить минимальный порядок фильтра при разработке устройства;

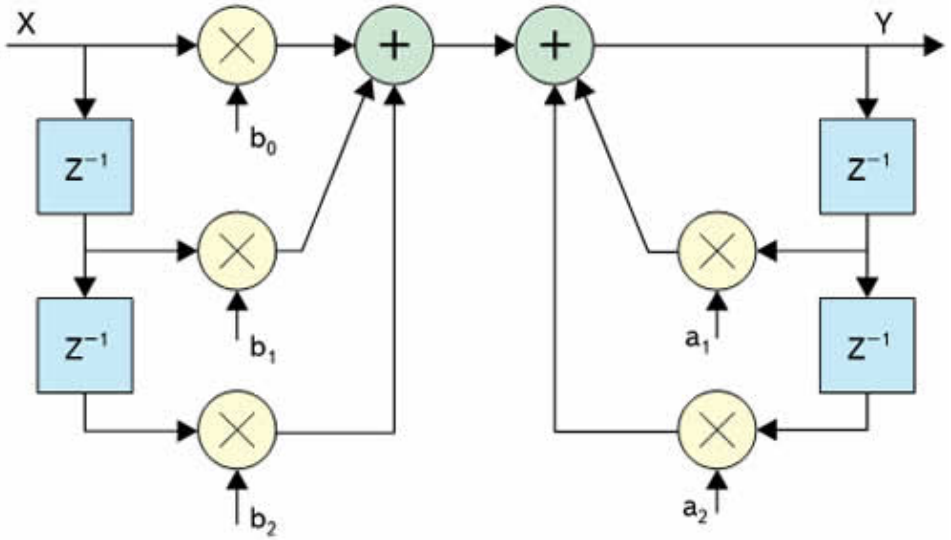
- реализовать фильтр на базе дискретных элементов, линий, ПЛИС или другого вычислителя.

Нужно решить задачу проектирования фильтра

- найти коэффициенты фильтра при заданных ограничениях на АЧХ/ФЧХ



⇔



a, b – ???

Задача проектирования

По виду цифрового фильтра:

- ищем в классе КИХ фильтров; $$H(z) = b_0 + b_1 z^{-1} + \ldots + b_N z^{-N}$$

- ищем в классе БИХ фильтров. $$H(z) = \frac{b_0 + b_1 z^{-1} + \ldots + b_N z^{-N}}{1 + a_1 z^{-1} + \ldots + a_M z^{-M}}$$

По наличию аналогового прототипа:

- методы синтеза с использование аналогового прототипа (лекция 7, билинейное преобразование, метод замены дифференциалов, метод инвариантной импульсной характеристики)

-прямые методы синтеза по ограничениям на АЧХ, ФЧХ, ПФ, ИХ и т.д.

Прямые методы:
- оптимальные (ищем точное соответствие критерию)
- субоптимальные (при дополнительных ограничениях на ПФ)

Аналоговые прототипы

Типы:

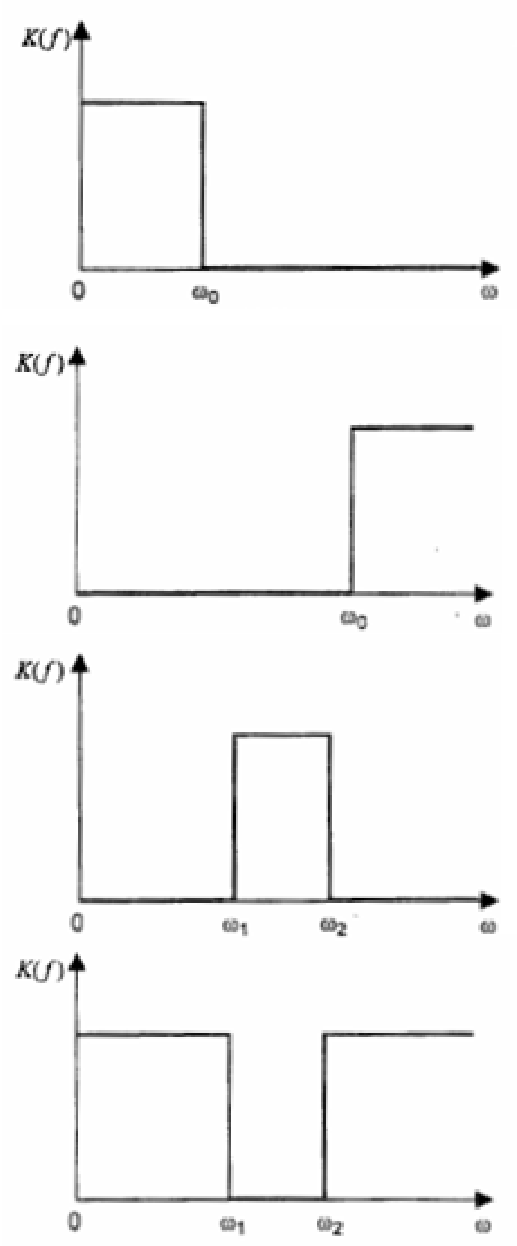
- фильтры нижних частот (ФНЧ, low-pass filter);

-фильтры верхних частот (ФВЧ, high-pass filter);

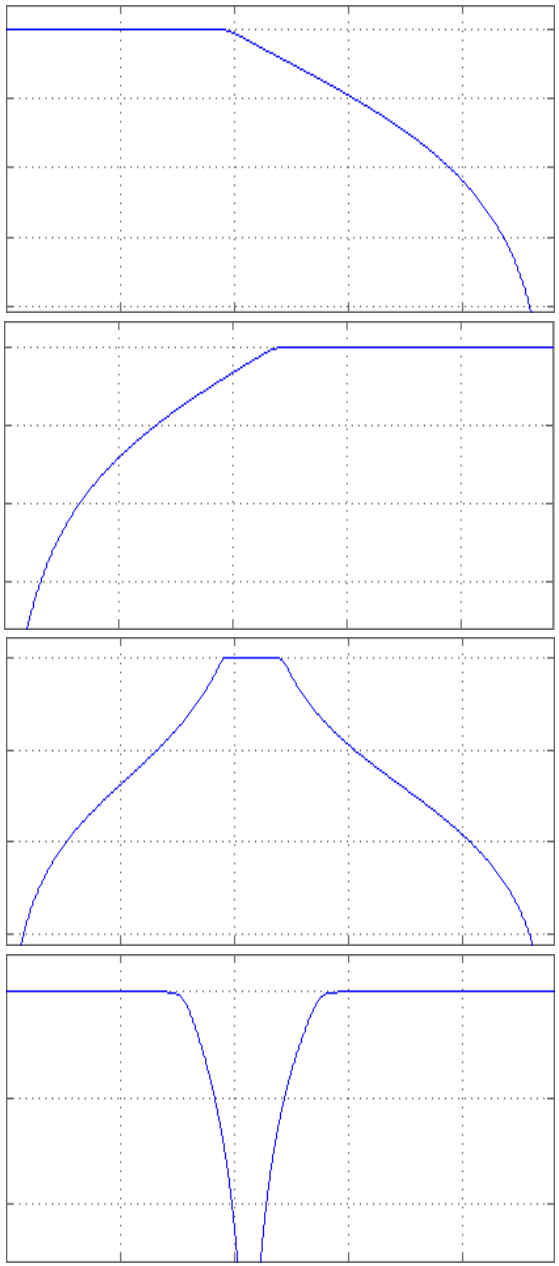
полосовые фильтры (band-pass filter);

- режекторные фильтры (band-stop filter).

Ожидание



Реальность

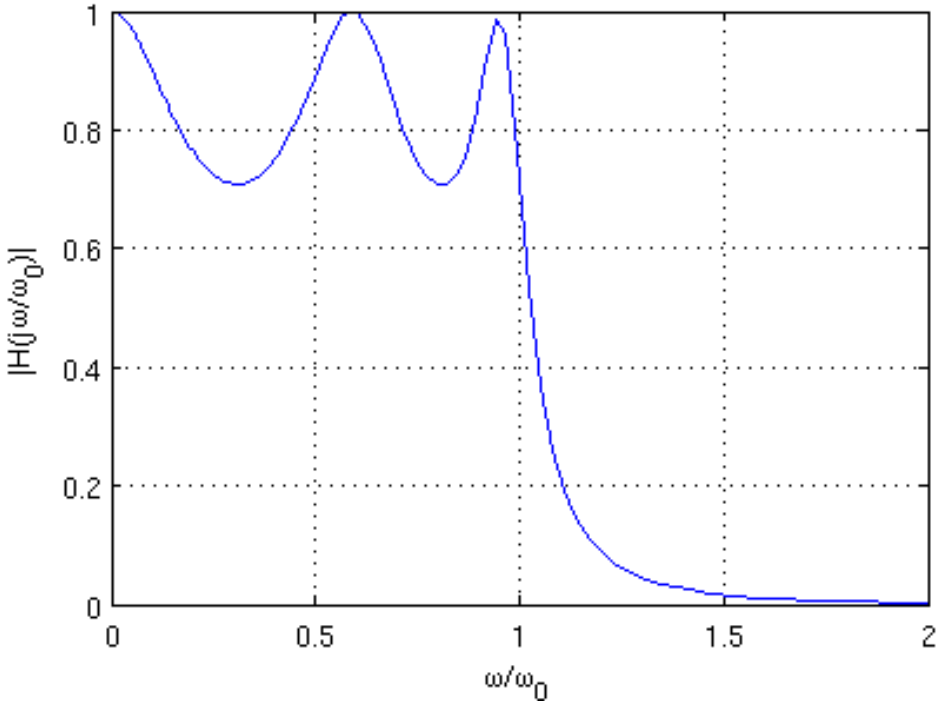


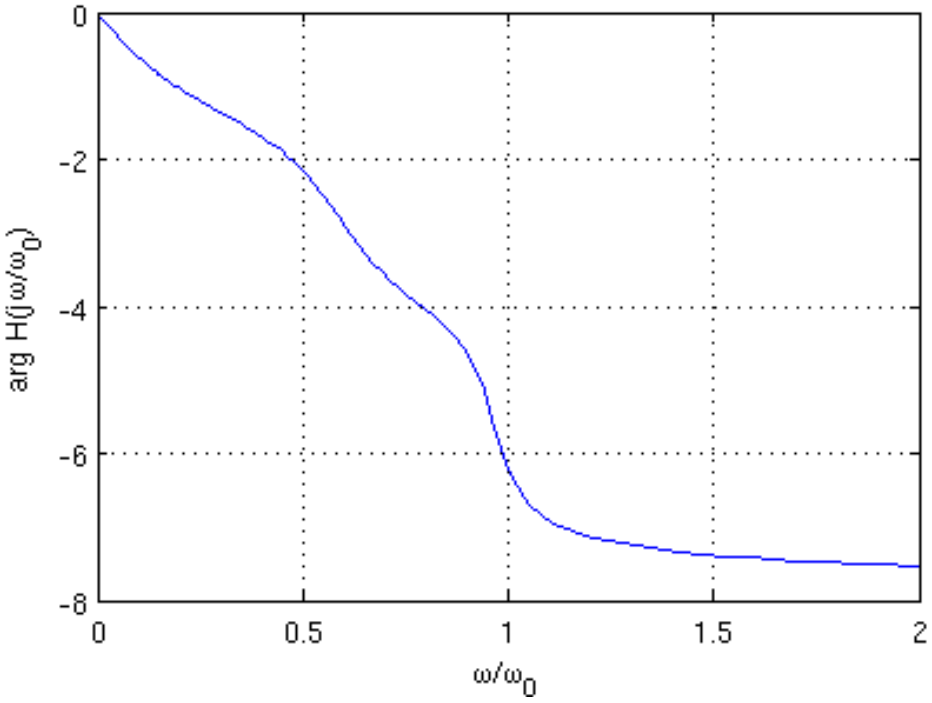
Аналоговые прототипы

Есть методы, аппроксимирующие идеальные передаточные функции.

Применим их для создания аналогового прототипа ФНЧ с нормированной частотой среза 1 рад/с

О том, как преобразовать один тип фильтра в другой (например, ФНЧ на ФВЧ) и сменить частоту среза, обсудим позднее.





Фильтр Баттерворта

Butterworth filter: максимальная гладкость

АЧХ:
(n – порядок)

$$K(\omega) = \frac{1}{\sqrt{1 + \left(\frac{\omega}{\omega_0}\right)^{2n}}}$$

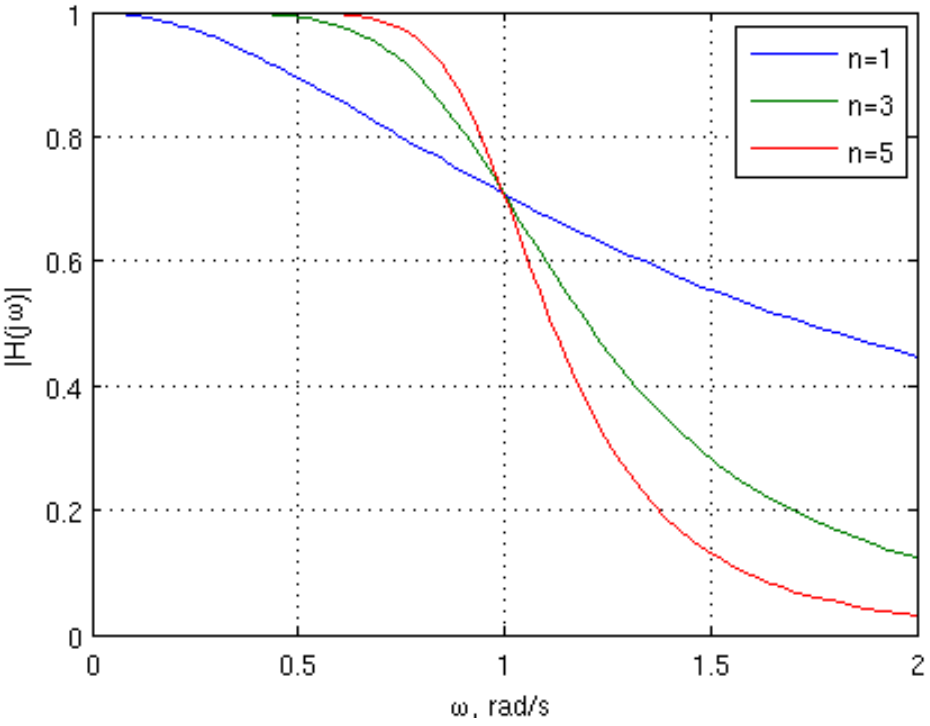
В нуле – единица,
на частоте среза – 0.707 (-3 дБ),
в бесконечности – ноль.

Монотонное спадание.

В нуле и бесконечности 2n-1 производных АЧХ равны нулю

```
clear all; clc; close all
j = 0;
for n = [1 3 5]
    j = j + 1;
    [z, p, k] = buttap(n);

    [b, a] = zp2tf(z, p, k);
    [Hn, wn] = freqs(b, a);
    H(:, j) = Hn; w(:, j) = wn;
end
plot(w, abs(H));
xlabel('\omega, rad/s');
ylabel('|H(j\omega)|');
legend('n=1', 'n=3', 'n=5');
grid on
```



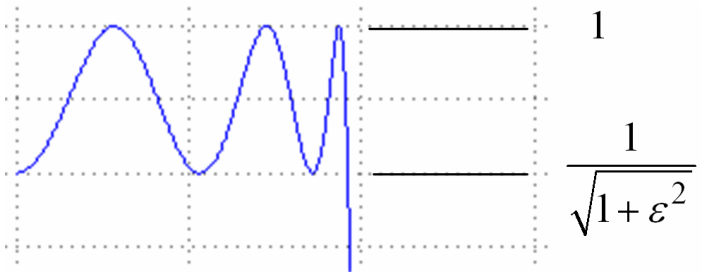
Фильтр Чебышева 1 рода

Chebyshev type I filter: резкий спад

АЧХ: $$K(\omega)=\frac{1}{\sqrt{1+\varepsilon^2 T_n\left(\omega/\omega_0\right)}}$$

$T_n(\ )$ **- полином Чебышева**

ε **определяет пульсации в полосе пропускания**



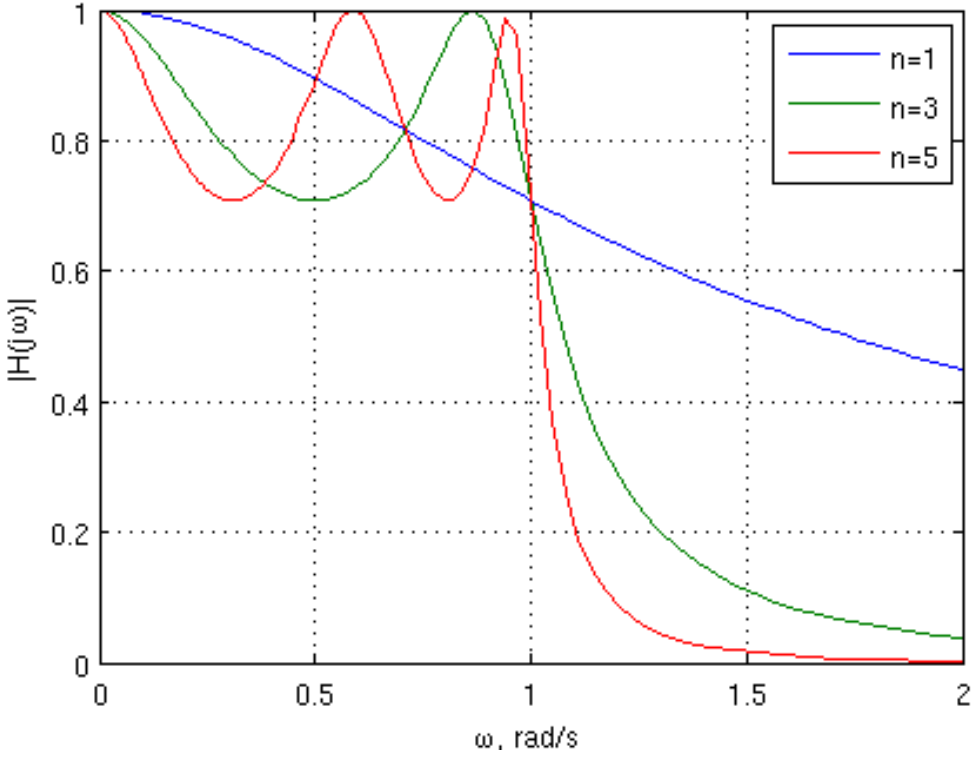
Монотонно затухает к нулю после частоты среза

В нуле 1 при нечетном n, иначе $\frac{1}{\sqrt{1+\varepsilon^2}}$

$$R_p = 20\log_{10}\left(\sqrt{1+\varepsilon^2}\right) =$$

$$= 10\log_{10}\left(1+\varepsilon^2\right)$$

```
...
Rp = 3;
for n = [1 3 5]
    j = j + 1;
    [z, p, k] = cheb1ap(n, Rp);
...
```



Фильтр Чебышева 2 рода

Chebyshev type II filter: резкий спад

АЧХ: $$K(\omega)=\frac{1}{\sqrt{1+\varepsilon^2 \Big/ T_n\left(\omega \Big/ \omega_0\right)}}$$

Связь с фильтром 1-го рода:

$$H_2(s)=1-H_1(1/s)$$

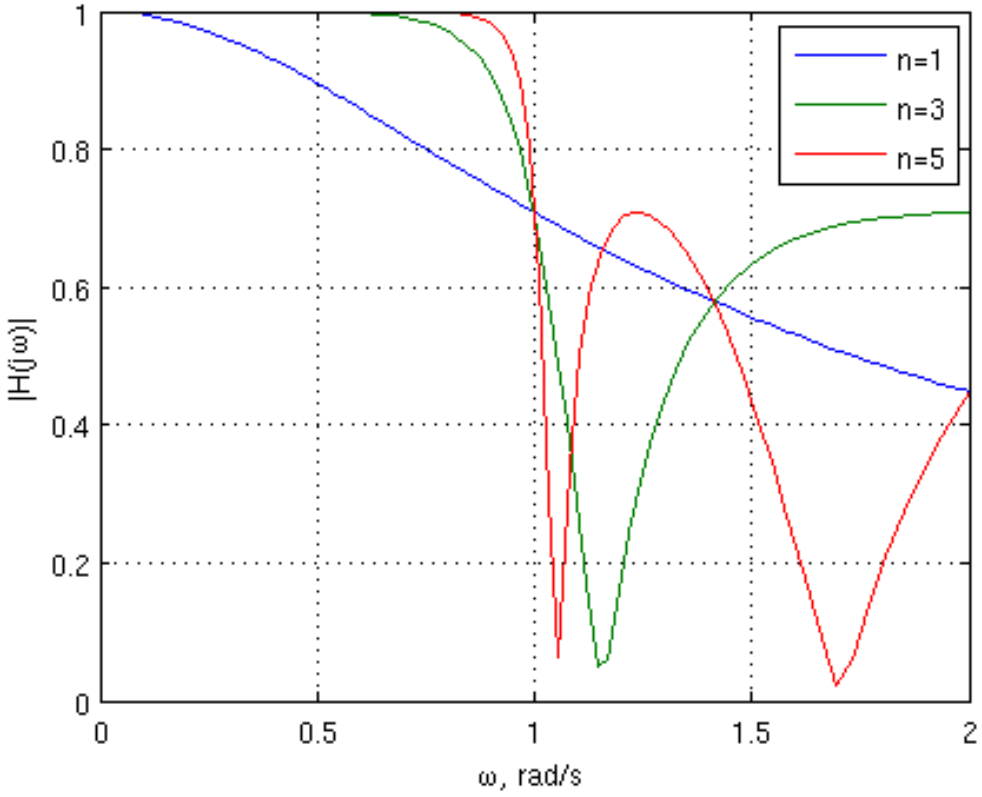
$$p_{2,i}=1 \Big/ p_{1,i}$$

Пульсации переносятся из полосы пропускания в полосу задержания,

там колеблются от 0 **до** $1 \Big/ \sqrt{1+\varepsilon^2}$

В нуле АЧХ максимально плоская

```
...
Rp = 3;
for n = [1 3 5]
    j = j + 1;
    [z, p, k] = cheb2ap(n, Rp);
...
```



Эллиптический фильтр

Elliptic filter, Cauer filter: фиксируем колебания, максимизируем крутизну

АЧХ: $$K(\omega)=\frac{1}{\sqrt{1+\varepsilon^2 R_n^2\left(\omega/\omega_0, L\right)}}$$

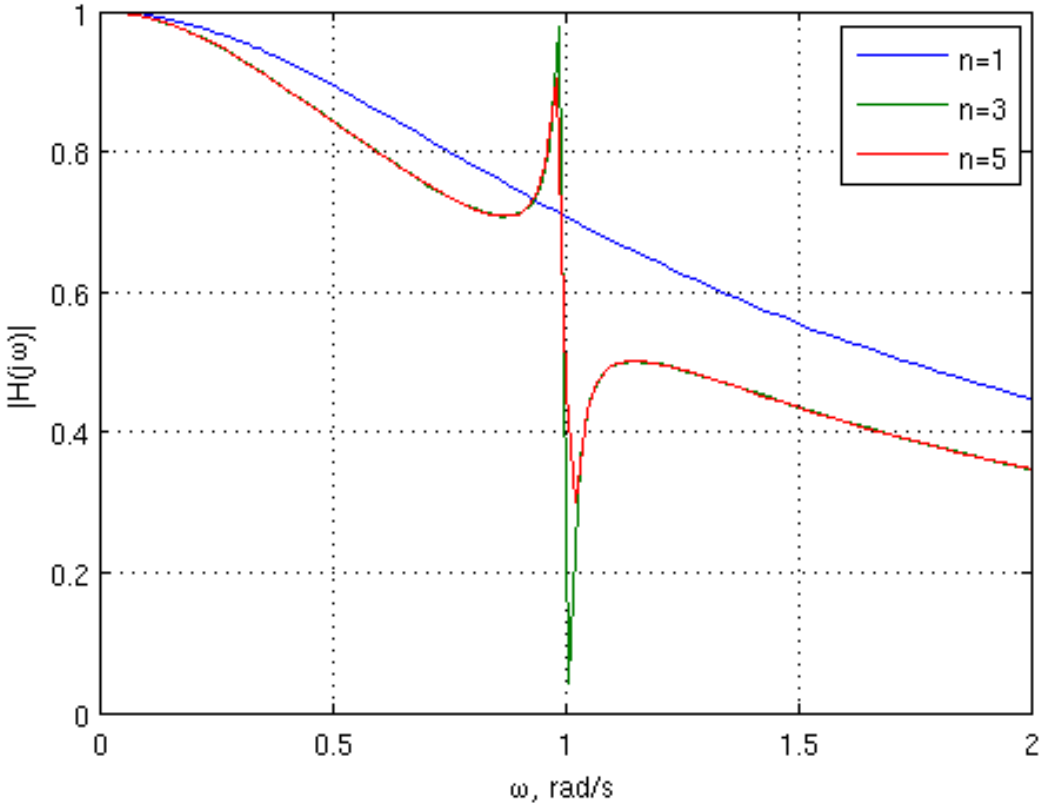
$R_n(\ )$ **- рациональная функция Чебышева**

ε, L **определяют пульсации в полосе пропускания и задержания**

Rp – пульсации в полосе пропускания (дБ),

Rs – пульсации в полосе задержания (дБ)

```
...
Rp = 3; Rs = 6;
for n = [1 3 5]
    j = j + 1;
    [z, p, k] = ellipap(n, Rp, Rs);
...
```



Фильтр Бесселя

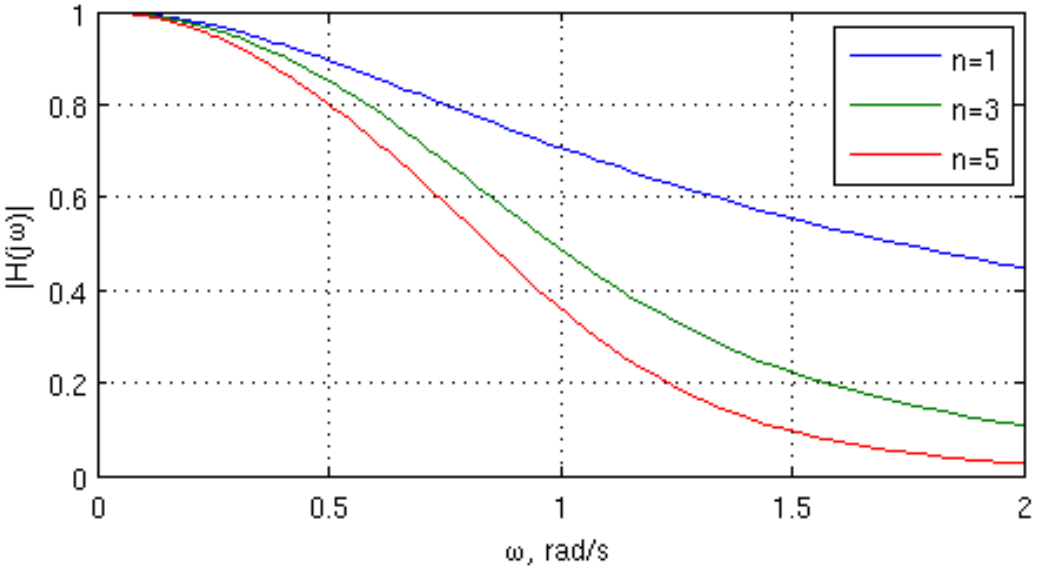
Bessel filter: гладкая ФЧХ

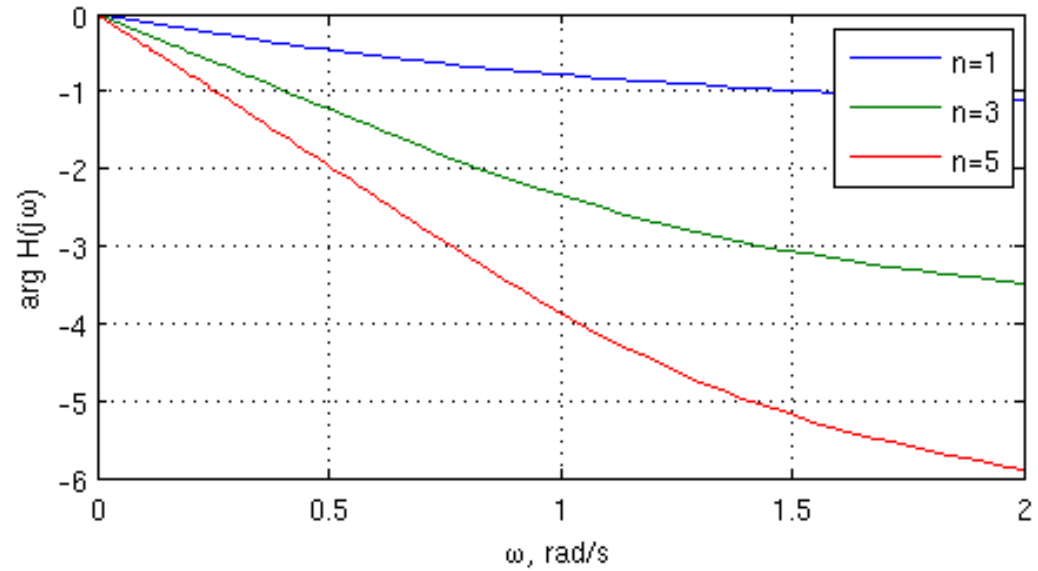
$$H(s) = \frac{d_0}{\sum_{k=0}^{n} d_k s^k}$$

$$d_k = \frac{(2n-k)!}{2^{n-k}\, k!\,(n-k)!}$$

```
clear all; clc; close all
j = 0;
for n = [1 3 5]
    j = j + 1;
    [z, p, k] = besselap(n);
...

subplot(2,1,2)
plot(w, unwrap(angle(H)));
xlabel('\omega, rad/s');
ylabel('arg H(j\omega)');
legend('n=1', 'n=3', 'n=5');
grid on; xlim([0 2]);
```





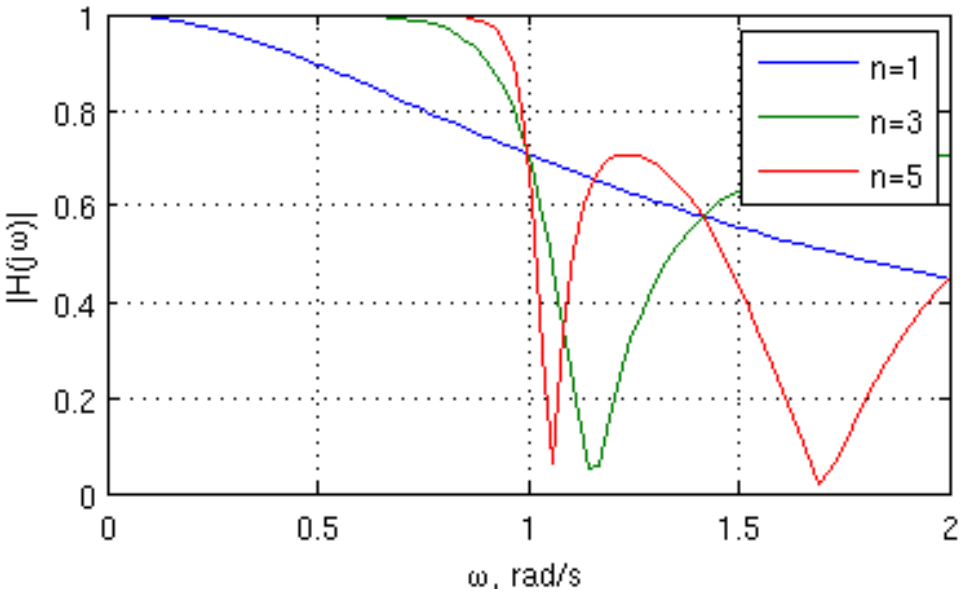
Смена частоты среза

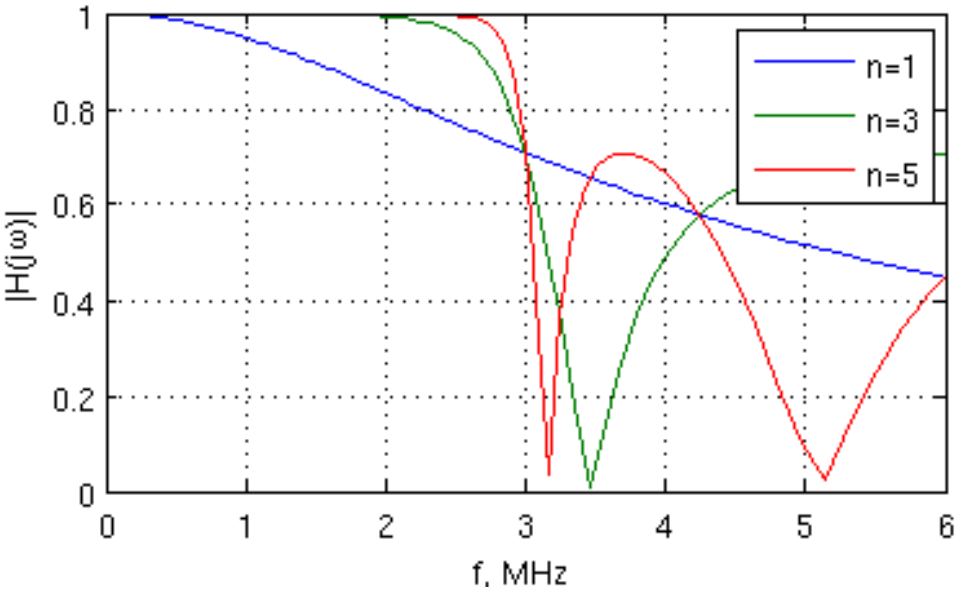
Смена частоты среза сводится к масштабированию частотной оси:

```
clear all; clc; close all
j = 0;
Rp = 3;
w0 = 2*pi*3e6;
for n = [1 3 5]
    j = j + 1;
    [z, p, k] = cheb2ap(n, Rp);
    [b, a] = zp2tf(z, p, k);

    [Hn, wn] = freqs(b, a);
    H(:, j) = Hn; w(:, j) = wn;

    [b_new, a_new] = lp2lp(b, a, w0);
    [Hn, wn] = freqs(b_new, a_new);
    H_new(:, j) = Hn; w_new(:, j) = wn;
end
figure(2)
plot(w_new/2/pi/1e6, abs(H_new));
xlabel('f, MHz');
ylabel('|H(j\omega)|');
legend('n=1', 'n=3', 'n=5');
grid on; xlim([0 6]);
```





Смена типа фильтра

Нелинейные преобразования оси частот позволяют сменить тип фильтра

ФНЧ в ФВЧ: $s \to \frac{\omega_0}{s}$

lp2hp

Transform lowpass analog filters to highpass

Syntax

```
[bt,at] = lp2hp(b,a,Wo)
[At,Bt,Ct,Dt] = lp2hp(A,B,C,D,Wo)
```

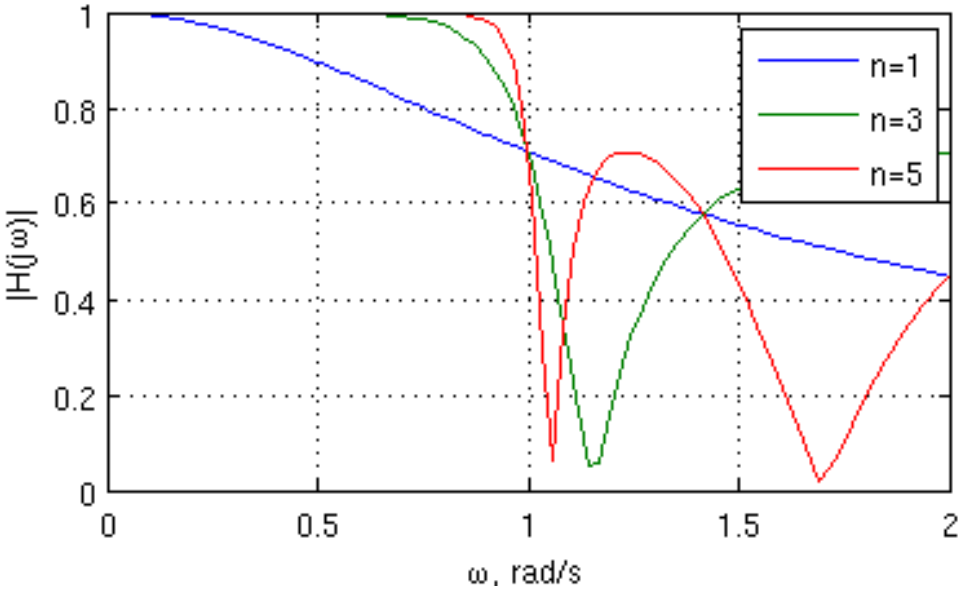
Description

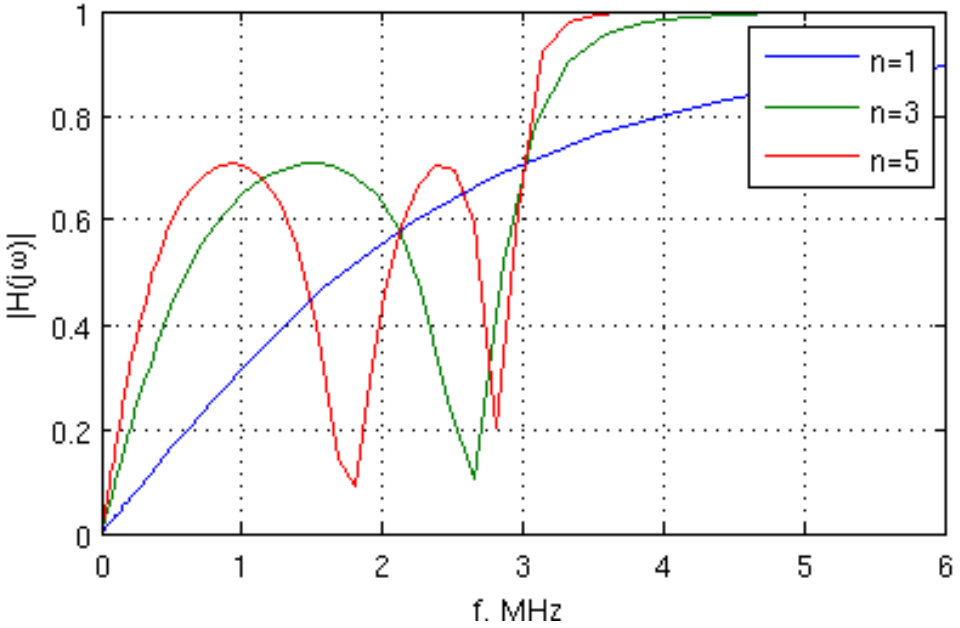
lp2hp transforms analog lowpass filter prototypes with a cutoff angular frequency of 1 rad/s into highpass filters with desired cutoff angular frequency. The transformation is one step in the digital filter design process for the butter, cheby1, cheby2, and ellip functions.

...

```
[b_new, a_new] = lp2hp(b, a, w0);
```

...





Смена типа фильтра

ФНЧ в ПФ:

$$s \to \frac{\omega_0}{B_w} \frac{\left(s/\omega_0\right)^2 + 1}{s/\omega_0}$$

lp2bp

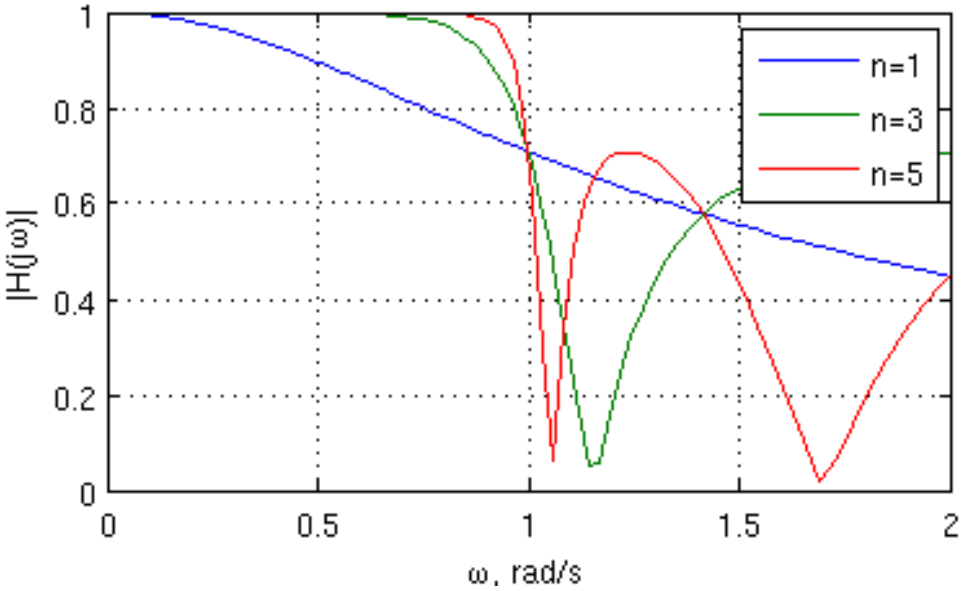
Transform lowpass analog filters to bandpass

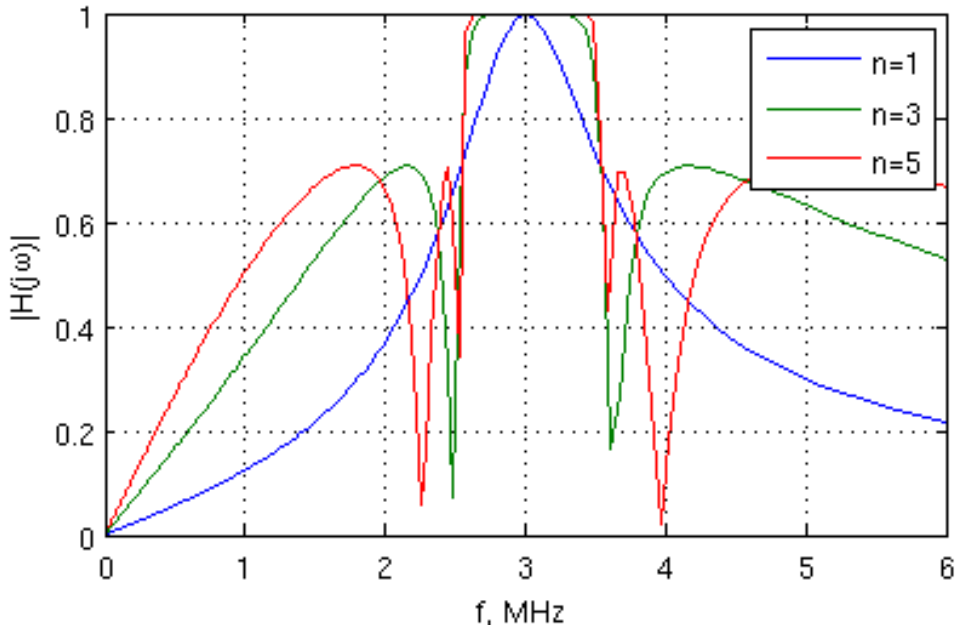
Syntax

```
[bt,at] = lp2bp(b,a,Wo,Bw)
[At,Bt,Ct,Dt] = lp2bp(A,B,C,D,Wo,Bw)
```

Description

lp2bp transforms analog lowpass filter prototypes with a cutoff angular frequency of 1 rad/s into bandpass filters with desired bandwidth and center frequency. The transformation is one step in the digital filter design process for the butter, cheby1, cheby2, and ellip functions.






```
Bw = 2*pi*1e6;
...
    [b_new, a_new] = lp2bp(b, a, w0, Bw);
...
```

Смена типа фильтра

ФНЧ в РФ:

$$s \to \frac{B_w}{\omega_0} \frac{s/\omega_0}{\left(s/\omega_0\right)^2 + 1}$$

lp2bs

Transform lowpass analog filters to bandstop

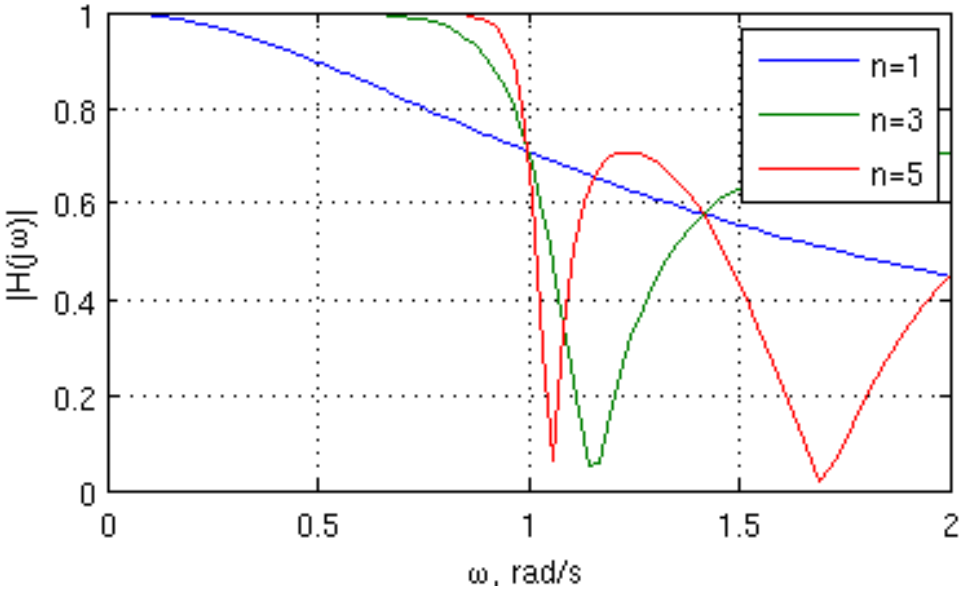
Syntax

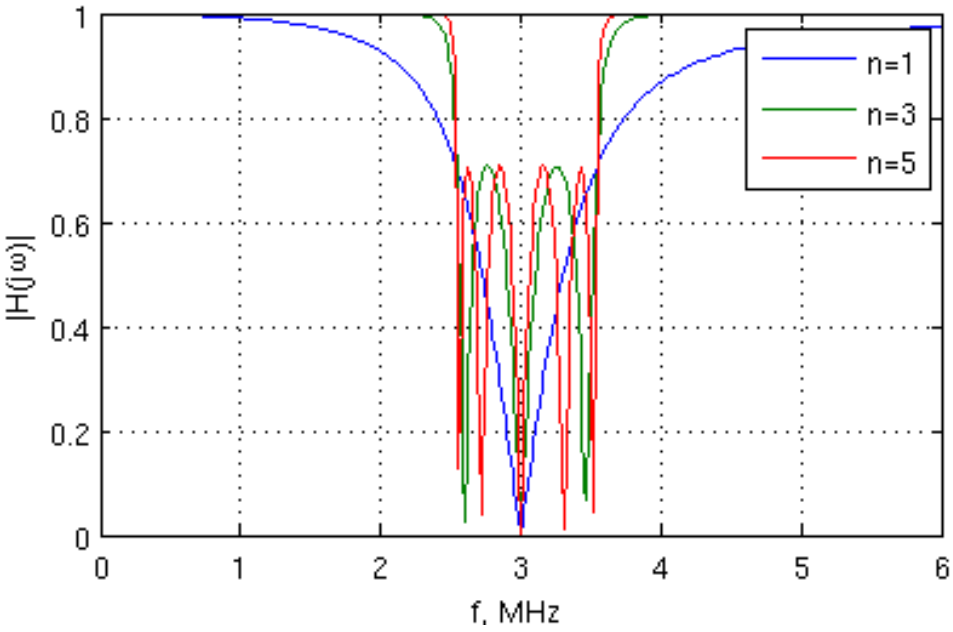
```
[bt,at] = lp2bs(b,a,Wo,Bw)
[At,Bt,Ct,Dt] = lp2bs(A,B,C,D,Wo,Bw)
```

Description

lp2bs transforms analog lowpass filter prototypes with a cutoff angular frequency of 1 rad/s into bandstop filters with desired bandwidth and center frequency. The transformation is one step in the digital filter design process for the butter, cheby1, cheby2, and ellip functions.

```
Bw = 2*pi*1e6;
…
    [b_new, a_new] = lp2bs(b, a, w0, Bw);
...
```





Функции расчета фильтров

В MATLAB есть функции-обертки, совмещающие расчет аналогового прототипа и преобразование его в требуемый тип фильтра:

butter

Butterworth filter design

Syntax

```
[z,p,k]=butter(n,Wn)
[z,p,k] = butter(n,Wn,'ftype')
[b,a]=butter(n,Wn)
[b,a]=butter(n,Wn,'ftype')
[A,B,C,D]=butter(n,Wn)
[A,B,C,D] = butter(n,Wn,'ftype')
[z,p,k]=butter(n,Wn,'s')
[z,p,k] = butter(n,Wn,'ftype','s')
[b,a]=butter(n,Wn,'s')
[b,a]=butter(n,Wn,'ftype','s')
```

ellip

Elliptic filter design

Syntax

```
[z,p,k]=ellip(n,Rp,Rs,Wp)
[z,p,k] = ellip(n,Rp,Rs,Wp,'ftype')
[b,a]=ellip(n,Rp,Rs,Wp)
[b,a]=ellip(n,Rp,Rs,Wp,'ftype')
[A,B,C,D]=ellip(n,Rp,Rs,Wp)
[A,B,C,D]=ellip(n,Rp,Rs,Wp,'ftype')
[z,p,k]=ellip(n,Rp,Rs,Wp,'s')
[z,p,k]=ellip(n,Rp,Rs,Wp,'ftype','s')
[b,a]=ellip(n,Rp,Rs,Wp,'s')
[b,a]=ellip(n,Rp,Rs,Wp,'ftype','s')
```

cheby1

Chebyshev Type I filter design (passband ripple)

Syntax

```
[z,p,k]=cheby1(n,R,Wp)
[z,p,k]=cheby1(n,R,Wp,'ftype')
[b,a]=cheby1(n,R,Wp)
[b,a]=cheby1(n,R,Wp,'ftype')
[A,B,C,D]=cheby1(n,R,Wp)
[A,B,C,D]=cheby1(n,R,Wp,'ftype')
[z,p,k]=cheby1(n,R,Wp,'s')
[z,p,k] = cheby1(n,R,Wp,'ftype','s')
[b,a]=cheby1(n,R,Wp,'s')
[b,a]=cheby1(n,R,Wp, 'ftype','s')
```

cheby2

Chebyshev Type II filter design (stopband ripple)

Syntax

```
[z,p,k]=cheby2(n,R,Wst)
[z,p,k]=cheby2(n,R,Wst,'ftype')
[b,a]=cheby2(n,R,Wst)
[b,a]=cheby2(n,R,Wst,'ftype')
[A,B,C,D]=cheby2(n,R,Wst)
[A,B,C,D]=cheby2(n,R,Wst,'ftype')
[z,p,k]=cheby2(n,R,Wst,'s')
[z,p,k]=cheby2(n,R,Wst,'ftype','s')
[b,a]=cheby2(n,R,Wst,'s')
[b,a]=cheby2(n,R,Wst,'ftype','s')
```

Переход к ДФ

Получили аналоговый прототип – можем переходить к дискретному фильтру с помощью:
- метода инвариантной импульсной характеристики;
- метода билинейного преобразования;
- метод замены дифференциалов.

А можем функции *butter*, *ellip*, *cheby1*, *cheby2* сразу вызвать без ключа ‘s’, тогда MATLAB применит билинейное преобразование к прототипу.

bilinear

Bilinear transformation method for analog-to-digital filter conversion

Syntax

```
[zd,pd,kd]=bilinear(z,p,k,fs)
[zd,pd,kd]=bilinear(z,p,k,fs,fp)
[numd,dend]=bilinear(num,den,fs)
[numd,dend]=bilinear(num,den,fs,fp)
[Ad,Bd,Cd,Dd]=bilinear(A,B,C,D,fs)
[Ad,Bd,Cd,Dd]=bilinear(A,B,C,D,fs,fp)
```

Description

The *bilinear transformation* is a mathematical mapping of variables. In digital filtering, it is a standard method of mapping the *s* or analog plane into the *z* or digital plane. It transforms analog filters, designed using classical filter design techniques, into their discrete equivalents.

impinvar

Impulse invariance method for analog-to-digital filter conversion

Syntax

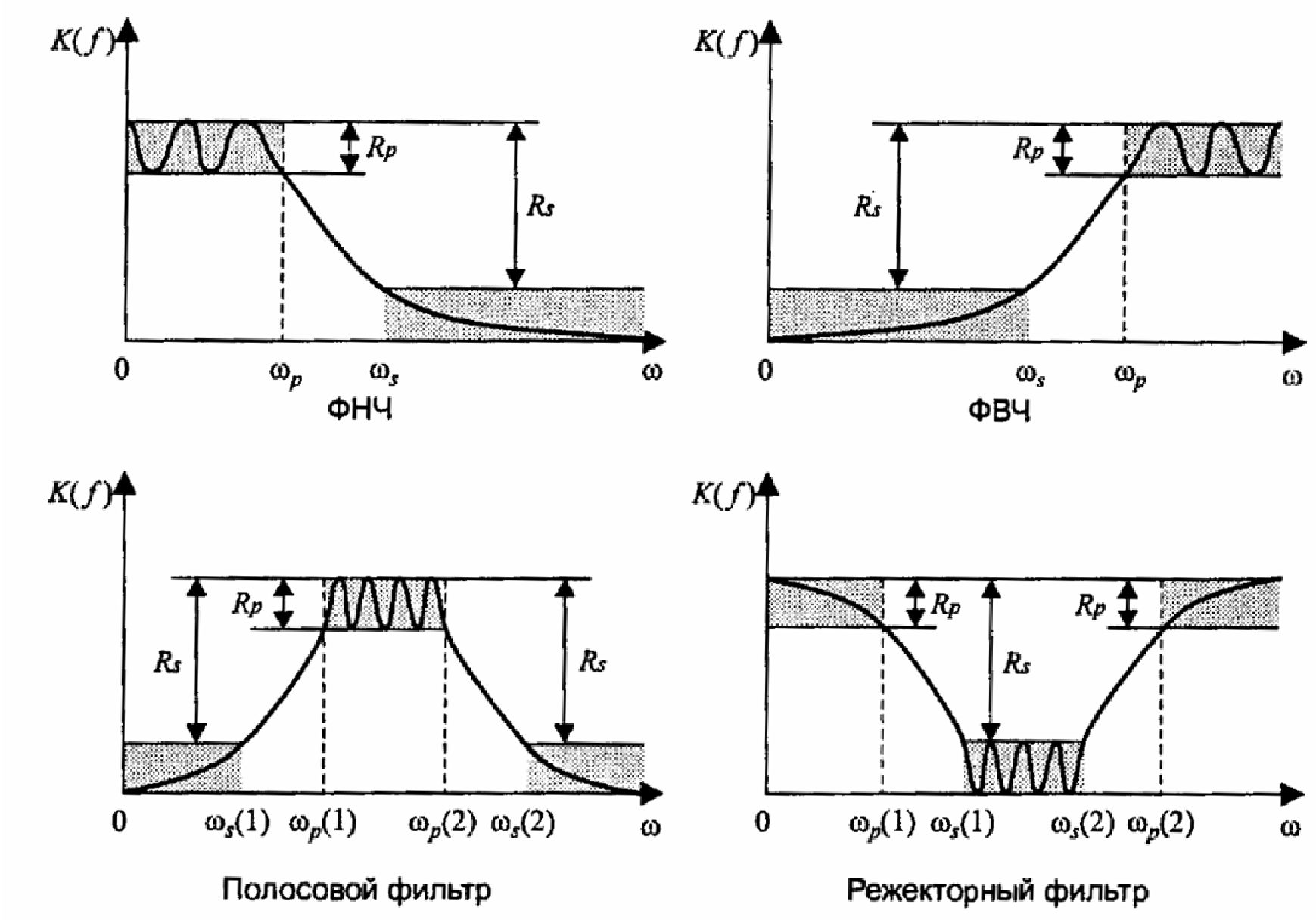
```
[bz,az] = impinvar(b,a,fs)
[bz,az] = impinvar(b,a,fs,tol)
```

Description

`[bz,az] = impinvar(b,a,fs)` creates a digital filter with numerator and denominator coefficients bz and az, respectively, whose impulse response is equal to the impulse response of the analog filter with coefficients b and a, scaled by 1/fs. If you leave out the argument fs, or specify fs as the empty vector `[]`, it takes the default value of 1 Hz.

Порядок фильтра

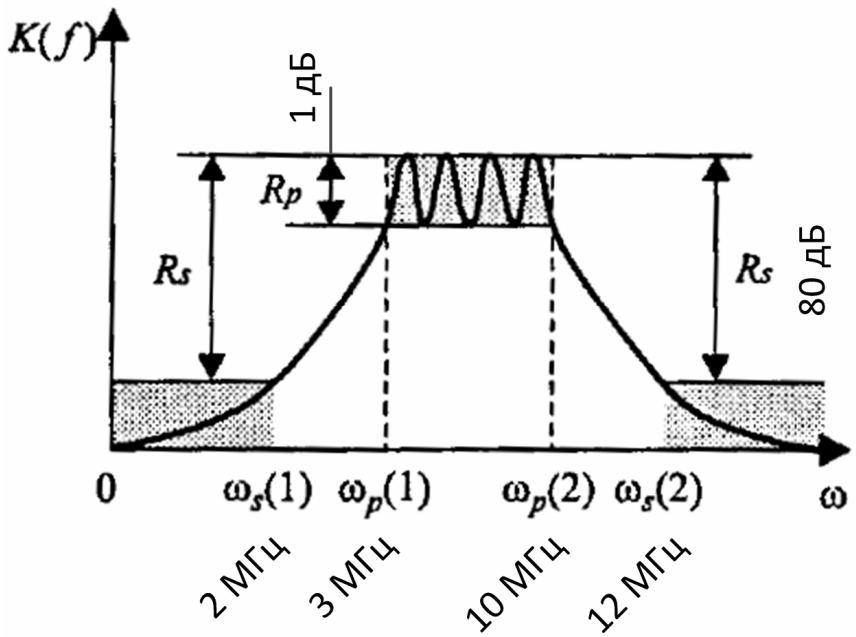
Допустим, с типом фильтра определились, нужно выбрать порядок.



Порядок фильтра

```
[n, Wn] = buttord(Wp, Ws, Rp, Rs, 's')
[n, Wn] = cheb1ord(Wp, Ws, Rp, Rs, 's')
[n, Wn] = cheb2ord(Wp, Ws, Rp, Rs, 's')
[n, Wn] = ellipord(Wp, Ws, Rp, Rs, 's')
```

n **– порядок,** Wn **– частота среза**

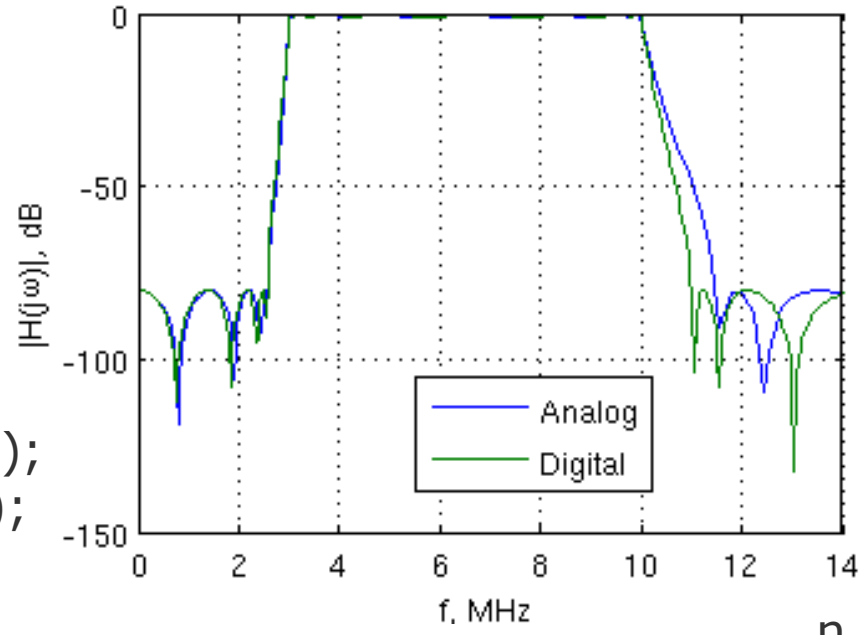



```
clear all; clc; close all
Rp = 1; Rs = 80;
Ws = [2 12] * 1e6 * 2*pi;
Wp = [3 10] * 1e6 * 2*pi;

[n, Wn] = ellipord(Wp, Ws, Rp, Rs, 's');
[bs, as] = ellip(n, Rp, Rs, Wp, 's');
Wd = 40 * 1e6 * 2*pi;
[bz, az] = ellip(n, Rp, Rs, Wp/(Wd/2));

[Hs, ws] = freqs(bs, as);
[Hz, wz] = freqz(bz, az);

plot(ws/2/pi/1e6, 20*log10(abs(Hs)), ...
 wz*Wd/(2*pi)^2/1e6, 20*log10(abs(Hz)));
xlabel('f, MHz'); ylabel('|H(j\omega)|, dB');
legend('Analog', 'Digital');
grid on; xlim([0 14])
```



n = 8

Оптимальный синтез

«Оптимальности» всегда соответствует критерий. Обычно задаются критерием минимизации нормы ошибок типа

$$L_p(e) = \int_{\omega_1}^{\omega_2} w(\omega)\left|D(\omega) - \left|H(j\omega)\right|\right|^p d\omega$$

где $D(\omega)$ **- требуемая АЧХ,**

$H(j\omega)$ **- искомая передаточная функция,**

$w(\omega)$ **- весовая функция.**

В общем случае решается итерационными численными методами.

При $p = 2$ **и поиске среди КИХ фильтров можно решить аналитически.**

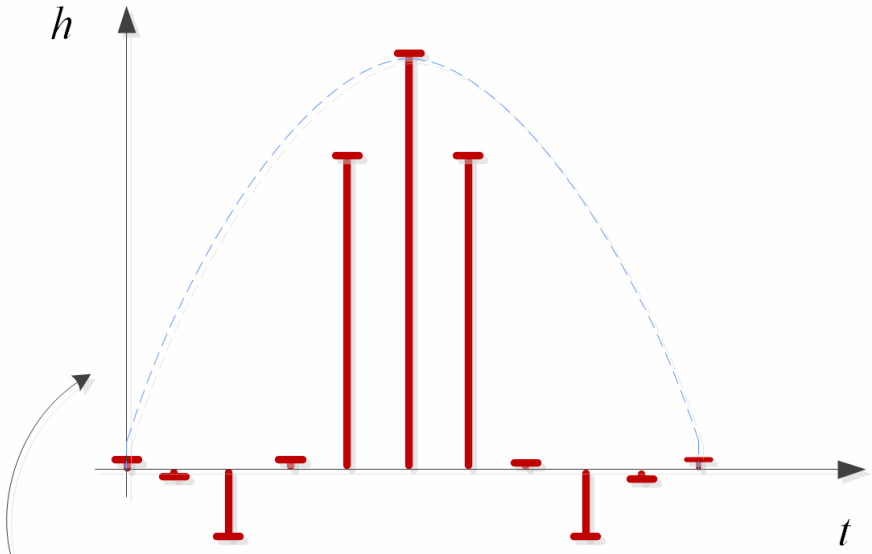
При $p = \infty$ **переходит в минимаксный критерий (минимизируем максимальное отклонение)**

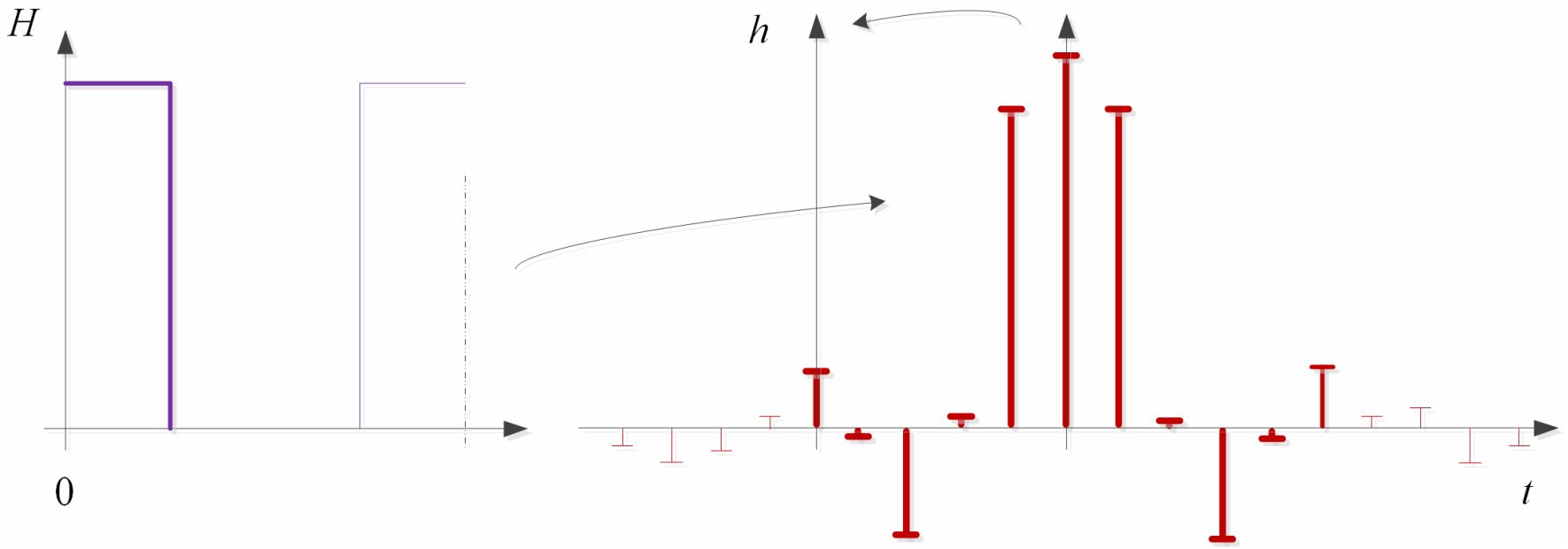
Субоптимальный синтез

Синтез с использованием окон:

1. АЧХ -> ИХ
2. Выбрать середину ИХ
3. Сдвинуть вправо
4. Наложить окно
5. …
6. Фильтр готов

Могут пригодиться функции:
infreqz(), prony(), fir1(), fir(2)





Субоптимальный синтез

Решение уравнений Юла-Уолкера (Yule-Walker):

yulewalk

Recursive digital filter design

Syntax

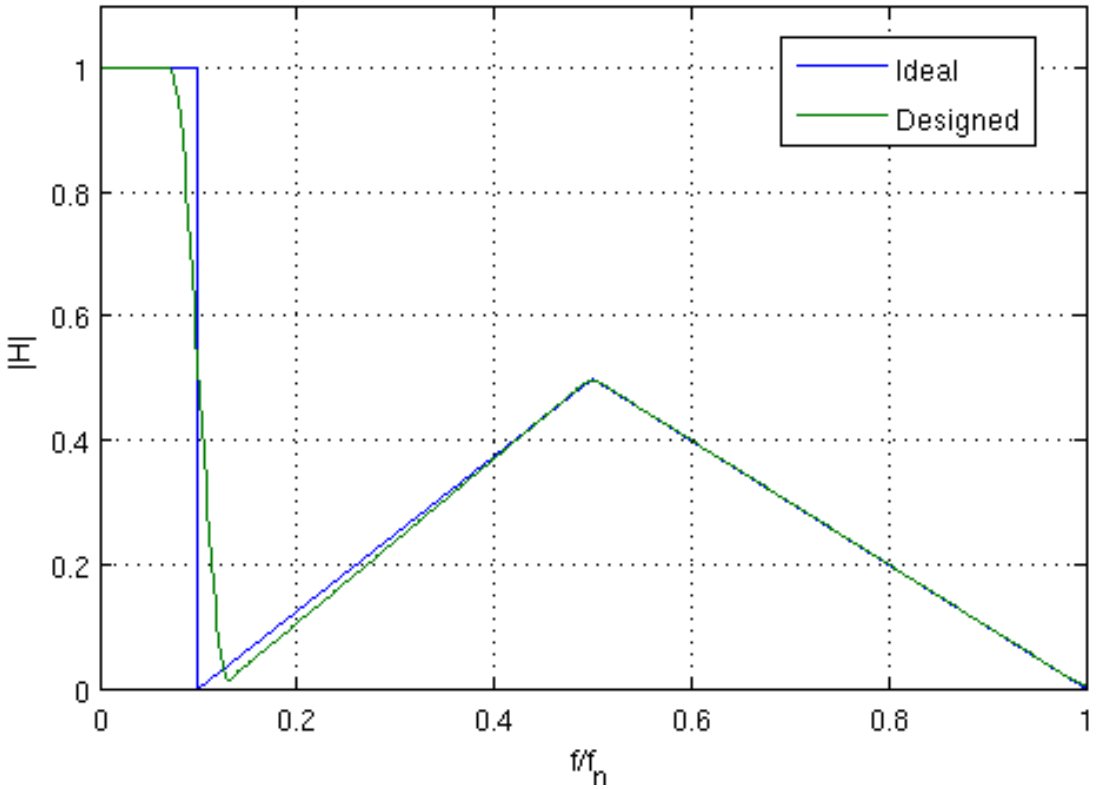
```
[b,a]=yulewalk(n,f,m)
```

Description

yulewalk designs recursive IIR digital filters using a least-squares fit to a specified frequency response.

`[b,a] = yulewalk(n,f,m)` returns row vectors b and a containing the n+1 coefficients of the order n IIR filter whose frequency-magnitude characteristics approximately match those given in vectors f and m:

Минимизирует среднеквадратическую ошибку для заданной кусочно-линейной АЧХ

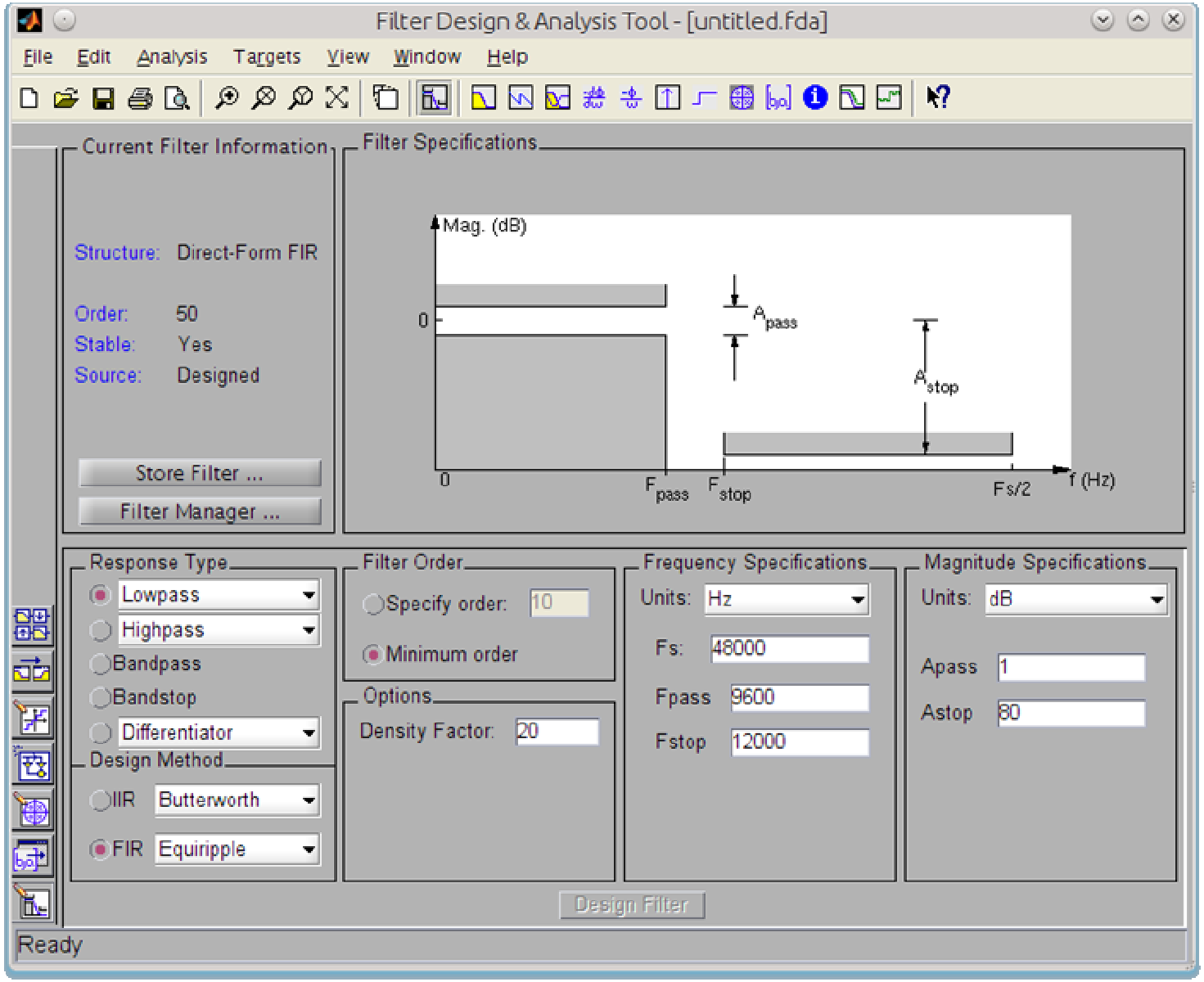



```
clear all; clc; close all;
f = [0 0.1 0.1 0.5 1 ];
m = [1 1 0 0.5 0];
n =30;
[b, a] = yulewalk(n, f, m);
[H, w] = freqz(b, a);
figure(1); plot(f, m, w/pi, abs(H));
xlabel('f/f_n'); ylabel('|H|');
ylim([0 1.1]); grid on; legend('Ideal', 'Designed');
```

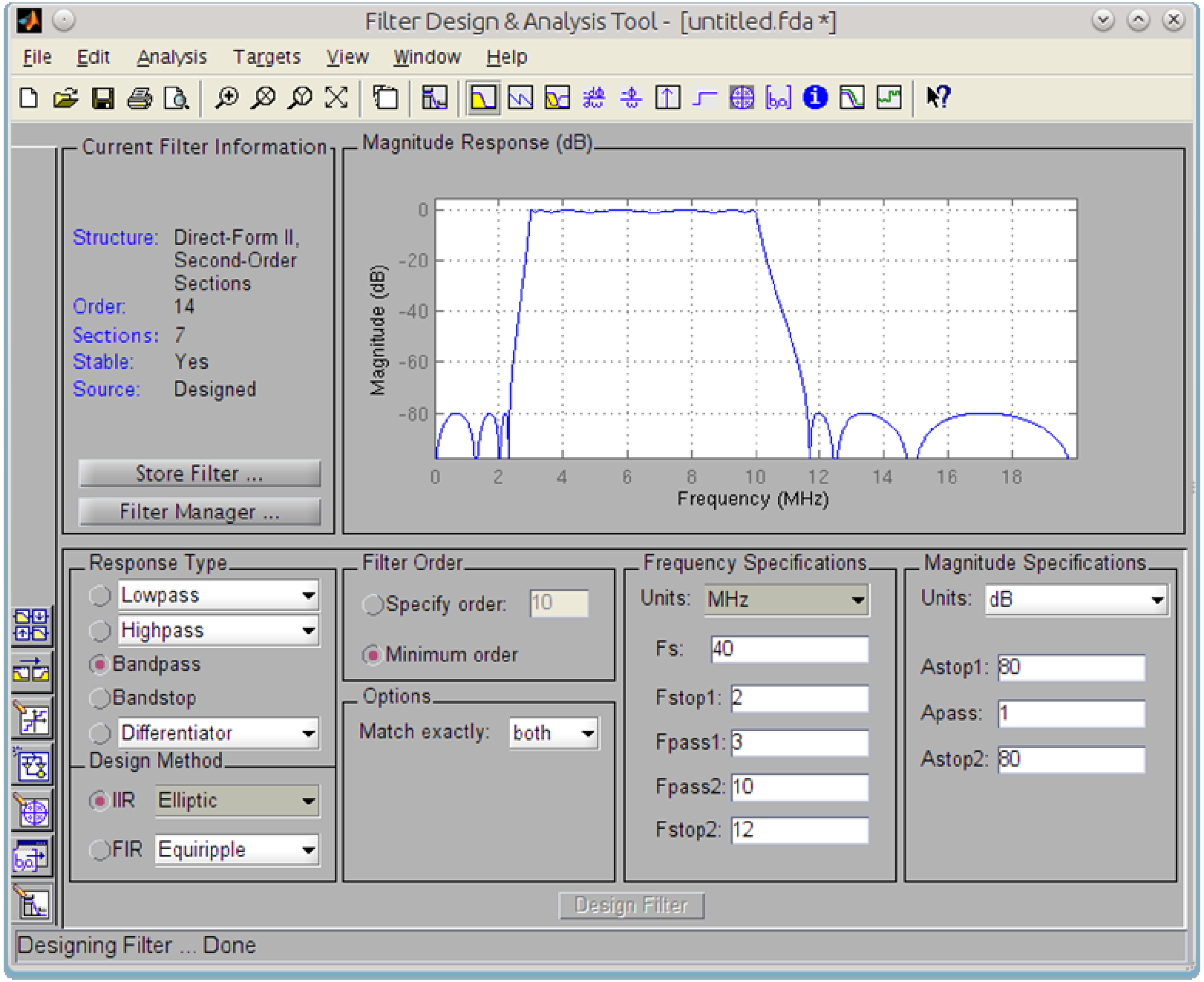
fdatool

Есть GUI, обобщающий изученные функции синтеза и анализа фильтров

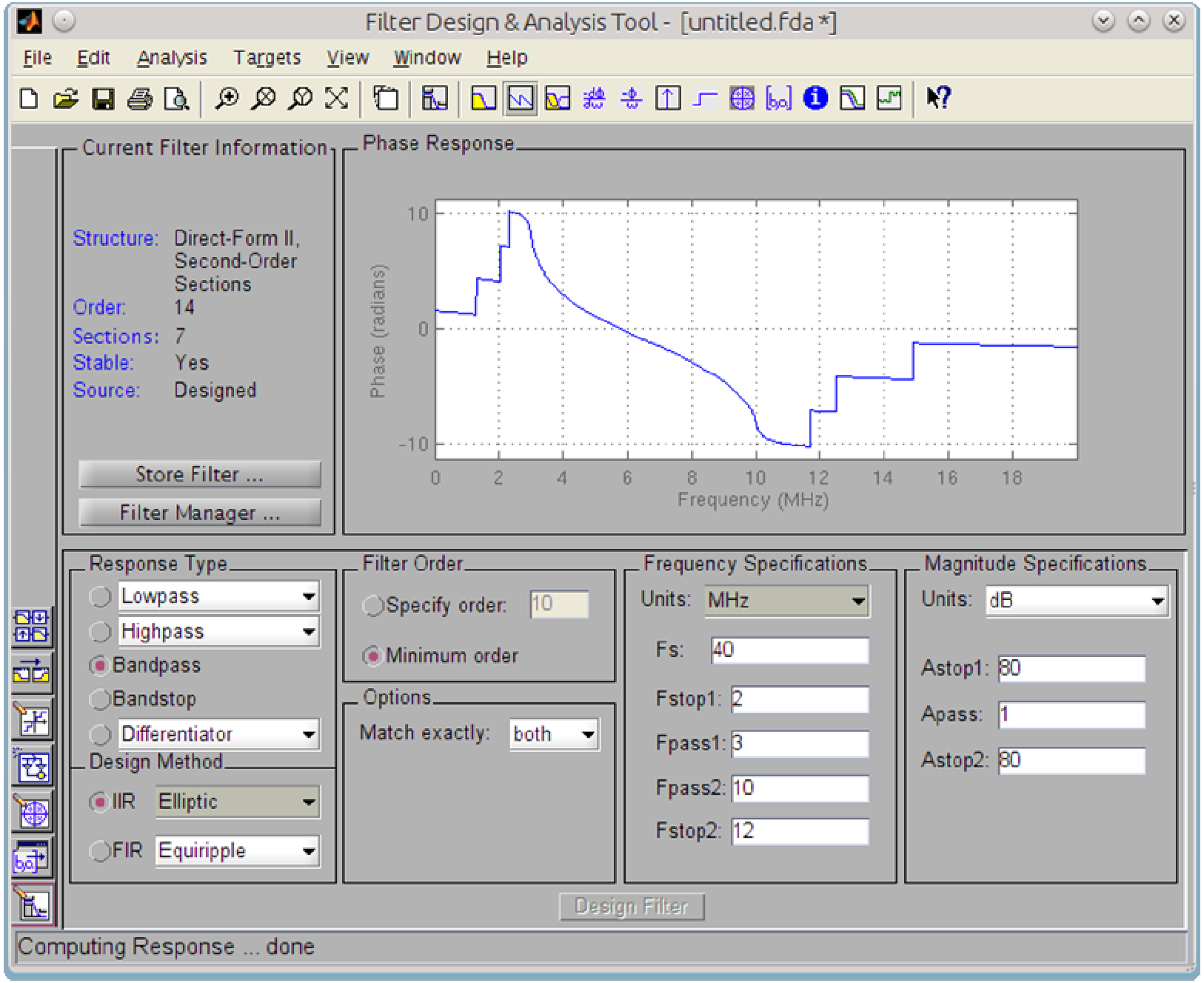
```
>> fdatool
```



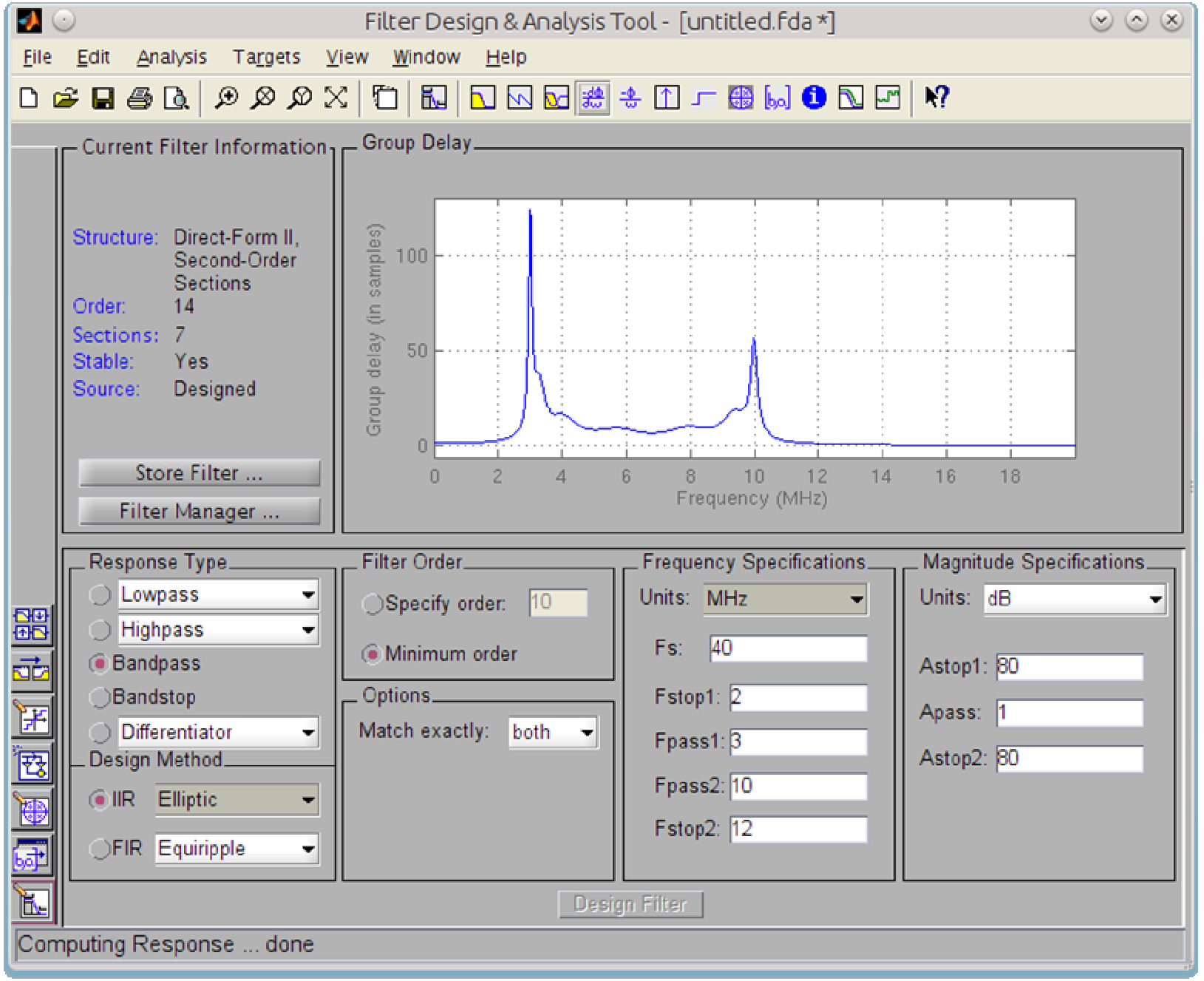
fdatool


Filter Design & Analysis Tool - [untitled.fda *]
File Edit Analysis Targets View Window Help
Current Filter Information
Structure: Direct-Form II, Second-Order Sections
Order: 14
Sections: 7
Stable: Yes
Source: Designed
Store Filter ...
Filter Manager ...
Magnitude Response (dB)
Magnitude (dB)
0 -20 -40 -60 -80
0 2 4 6 8 10 12 14 16 18
Frequency (MHz)
Response Type
Lowpass
Highpass
Bandpass
Bandstop
Differentiator
Design Method
IIR Elliptic
FIR Equiripple
Filter Order
Specify order: 10
Minimum order
Options
Match exactly: both
Frequency Specifications
Units: MHz
Fs: 40
Fstop1: 2
Fpass1: 3
Fpass2: 10
Fstop2: 12
Magnitude Specifications
Units: dB
Astop1: 80
Apass: 1
Astop2: 80
Design Filter
Designing Filter ... Done

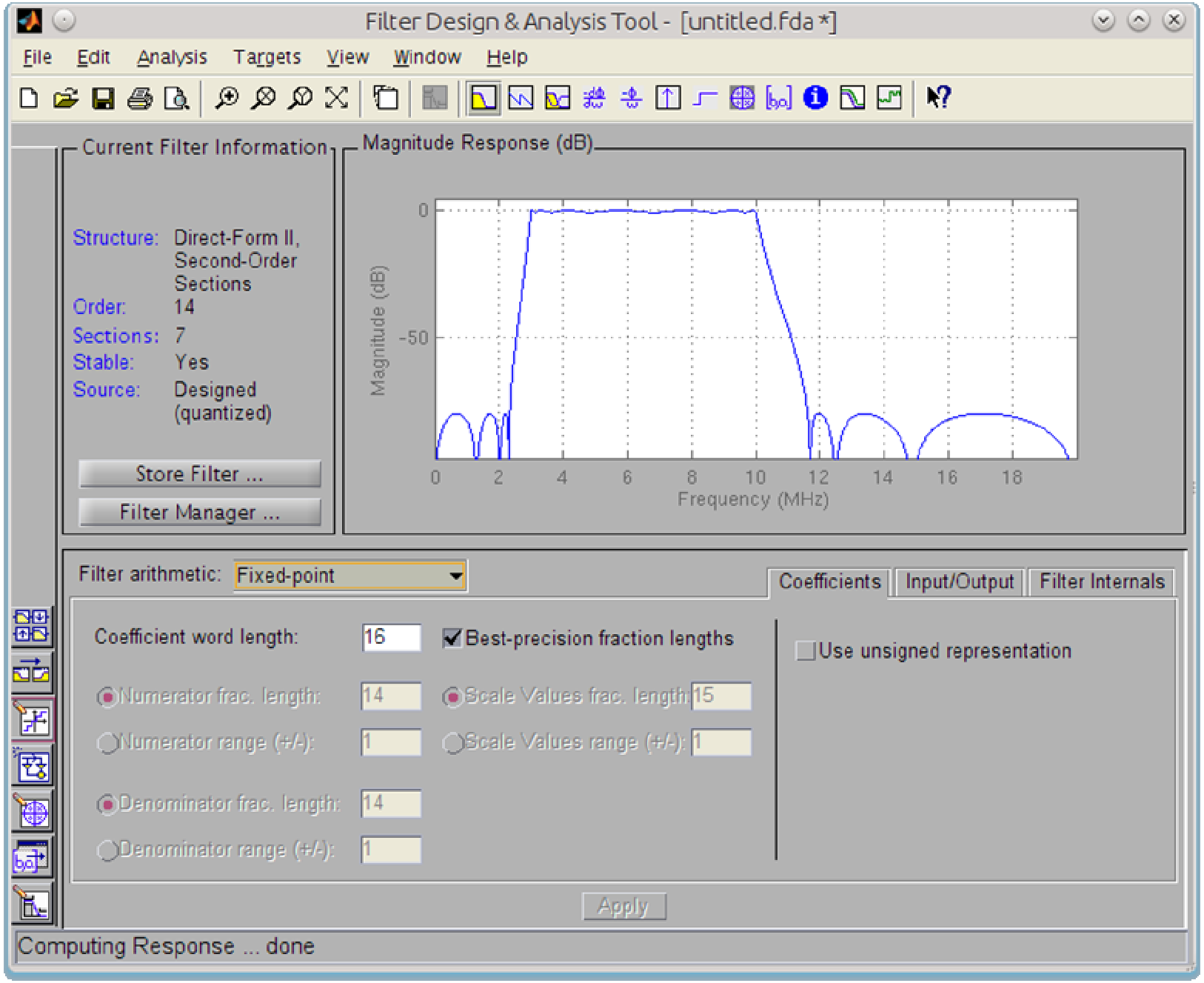
fdatool


Filter Design & Analysis Tool - [untitled.fda*]
File Edit Analysis Targets View Window Help
Current Filter Information
Structure: Direct-Form II, Second-Order Sections
Order: 14
Sections: 7
Stable: Yes
Source: Designed
Store Filter ...
Filter Manager ...
Phase Response
Phase (radians)
Frequency (MHz)
Response Type
Lowpass
Highpass
Bandpass
Bandstop
Differentiator
Design Method
IIR Elliptic
FIR Equiripple
Filter Order
Specify order: 10
Minimum order
Options
Match exactly: both
Frequency Specifications
Units: MHz
Fs: 40
Fstop1: 2
Fpass1: 3
Fpass2: 10
Fstop2: 12
Magnitude Specifications
Units: dB
Astop1: 80
Apass: 1
Astop2: 80
Design Filter
Computing Response ... done

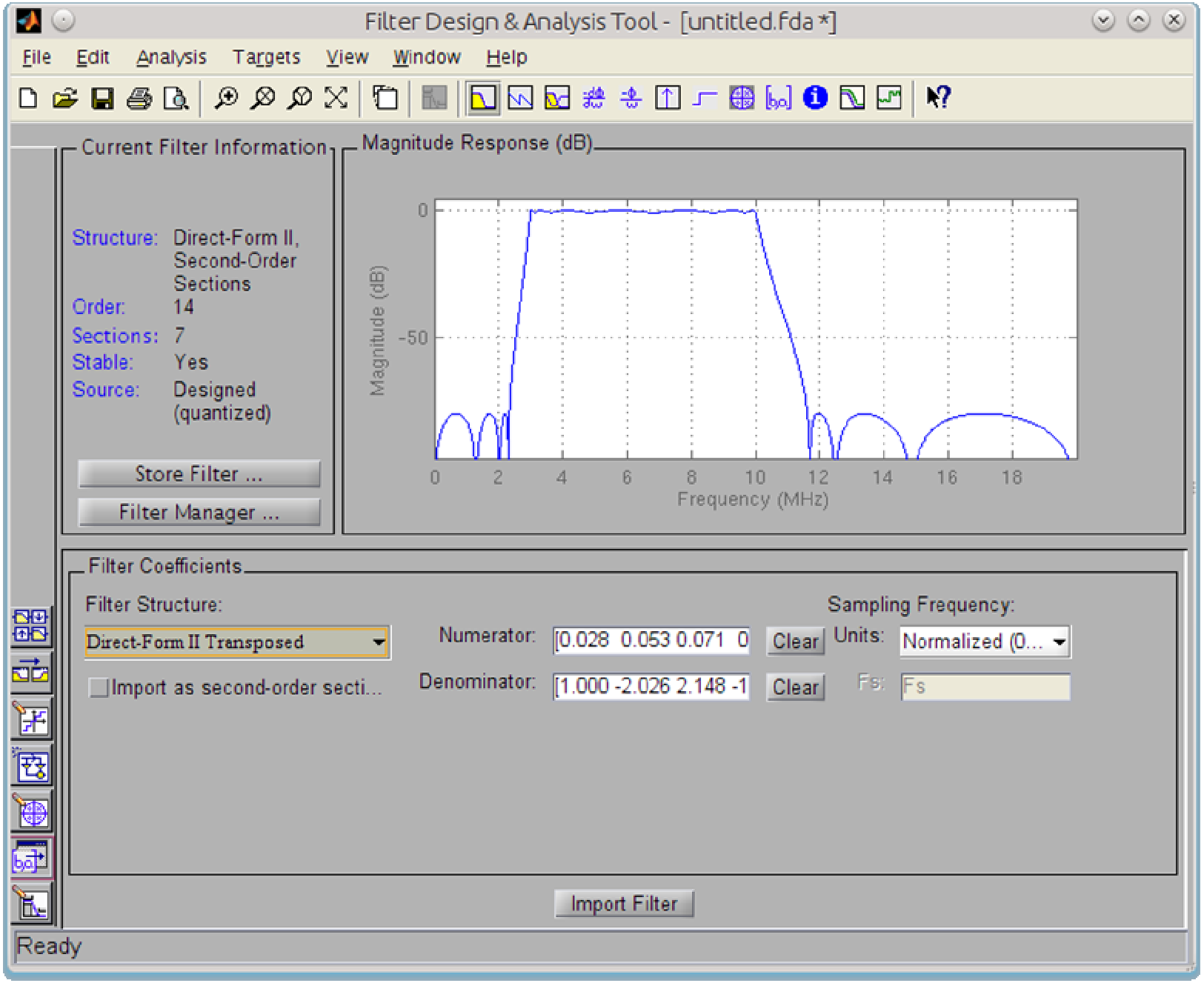
fdatool


Filter Design & Analysis Tool - [untitled.fda *]
File Edit Analysis Targets View Window Help
Current Filter Information
Structure: Direct-Form II, Second-Order Sections
Order: 14
Sections: 7
Stable: Yes
Source: Designed
Store Filter ...
Filter Manager ...
Group Delay
Group delay (in samples)
100
50
0
0 2 4 6 8 10 12 14 16 18
Frequency (MHz)
Response Type
Lowpass
Highpass
Bandpass
Bandstop
Differentiator
Design Method
IIR Elliptic
FIR Equiripple
Filter Order
Specify order: 10
Minimum order
Options
Match exactly: both
Frequency Specifications
Units: MHz
Fs: 40
Fstop1: 2
Fpass1: 3
Fpass2: 10
Fstop2: 12
Magnitude Specifications
Units: dB
Astop1: 80
Apass: 1
Astop2: 80
Design Filter
Computing Response ... done

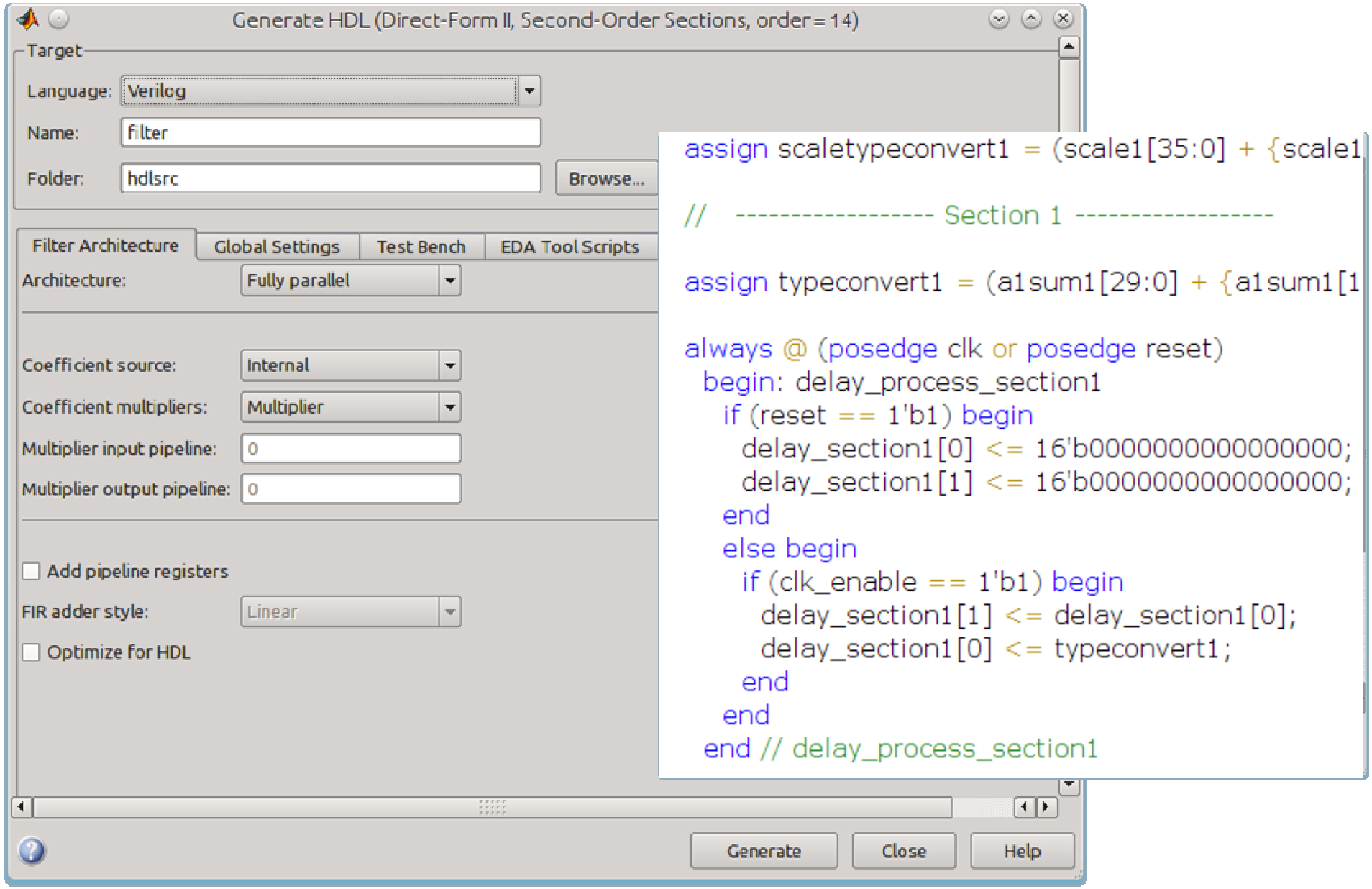
fdatool


Filter Design & Analysis Tool - [untitled.fda *]
File Edit Analysis Targets View Window Help
Current Filter Information
Structure: Direct-Form II, Second-Order Sections
Order: 14
Sections: 7
Stable: Yes
Source: Designed (quantized)
Store Filter ...
Filter Manager ...
Magnitude Response (dB)
Magnitude (dB)
0
-50
0 2 4 6 8 10 12 14 16 18
Frequency (MHz)
Filter arithmetic: Fixed-point
Coefficients
Input/Output
Filter Internals
Coefficient word length: 16
Best-precision fraction lengths
Use unsigned representation
Numerator frac. length: 14
Scale Values frac. length: 15
Numerator range (+/-): 1
Scale Values range (+/-): 1
Denominator frac. length: 14
Denominator range (+/-): 1
Apply
Computing Response ... done

fdatool


Filter Design & Analysis Tool - [untitled.fda*]
File Edit Analysis Targets View Window Help
Current Filter Information
Structure: Direct-Form II, Second-Order Sections
Order: 14
Sections: 7
Stable: Yes
Source: Designed (quantized)
Store Filter ...
Filter Manager ...
Magnitude Response (dB)
Magnitude (dB)
0
-50
0 2 4 6 8 10 12 14 16 18
Frequency (MHz)
Filter Coefficients
Filter Structure:
Direct-Form II Transposed
Import as second-order secti...
Numerator: [0.028 0.053 0.071 0
Denominator: [1.000 -2.026 2.148 -1
Clear
Clear
Sampling Frequency:
Units: Normalized (0...
Fs: Fs
Import Filter
Ready

fdatool


Generate HDL (Direct-Form II, Second-Order Sections, order = 14)
Target
Language: Verilog
Name: filter
Folder: hdlsrc
Browse...
Filter Architecture
Global Settings
Test Bench
EDA Tool Scripts
Architecture: Fully parallel
Coefficient source: Internal
Coefficient multipliers: Multiplier
Multiplier input pipeline: 0
Multiplier output pipeline: 0
Add pipeline registers
FIR adder style: Linear
Optimize for HDL
Generate
Close
Help
assign scaletypeconvert1 = (scale1[35:0] + {scale1
// ---------- Section 1 ----------
assign typeconvert1 = (a1sum1[29:0] + {a1sum1[1
always @ (posedge clk or posedge reset)
begin: delay_process_section1
if (reset == 1'b1) begin
delay_section1[0] <= 16'b0000000000000000;
delay_section1[1] <= 16'b0000000000000000;
end
else begin
if (clk_enable == 1'b1) begin
delay_section1[1] <= delay_section1[0];
delay_section1[0] <= typeconvert1;
end
end
end // delay_process_section1