Four-Pole Elliptical Low-pass Filter Datasheet ELPF4V 1.20

Copyright © 2009-2013 Cypress Semiconductor Corporation. All Rights Reserved.

Resources	PSoC® Blocks			API Memory (Bytes)		Pins (per External I/O)
	Digital	Analog CT	Analog SC	Flash ±3%	RAM	
CY8C29/28/24xxx, CY8C27x43						
	0	0	4	0	257	0

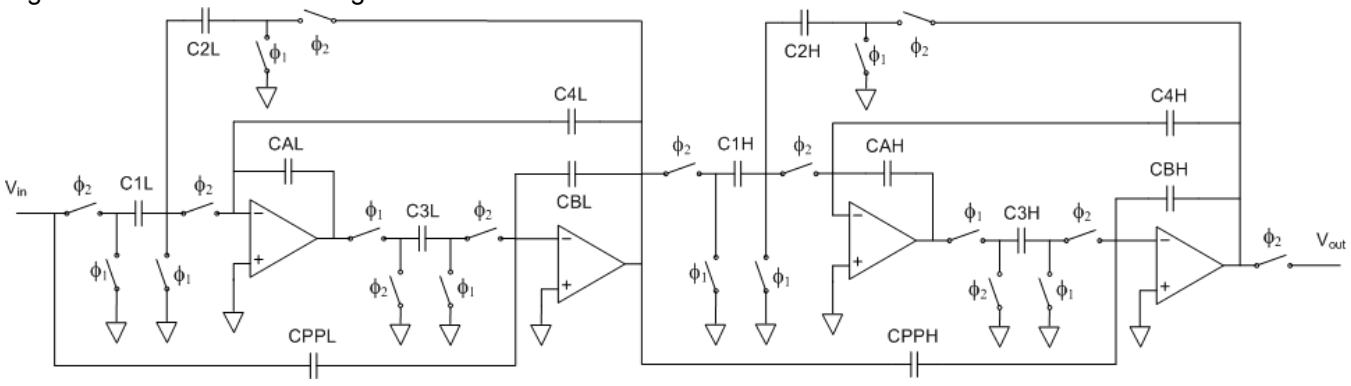
For one or more fully configured, functional example projects that use this user module go to www.cypress.com/psocexampleprojects

Features and Overview

- User-programmable gain
- User-programmable corner frequency (from 20 Hz to 150 kHz), notch frequency (for low and high filter pairs), and damping ratio with no external components
- Filter corner-frequency stability directly derived from clock accuracy
- Filter sampling rates up to 1.5 MHz

The ELPF4 User Module implements a general purpose four-pole low-pass filter with up to two pairs of elliptical zeroes at user selectable frequencies. The zeroes result in notches in the transfer function, intended to eliminate specific frequencies. Damping ratio are functions of the clock frequency and the ratios of the capacitor values calculated automatically based on the user parameters preset (Corner, Sample and Notch frequency, Gain, Filter type). Any of the classical all-pole filter configurations (Butterworth, Bessel, and Chebyshev) can be implemented. You can also select custom filter configuration, and enter values for complex poles. The corner frequency can be set very accurately or adjusted by controlling the sample rate clock. The filter output can drive the analog output bus.

Figure 1. ELPF4 Block Diagram



Functional Description

In the frequency domain, a four-pole low-pass filter has the frequency response:

$$\frac{V_{OUT}}{V_{IN}} = \frac{V_{OUT_L}}{V_{IN_L}} \times \frac{V_{OUT_H}}{V_{IN_H}}$$

The frequency response for Higher and Lower filter parts is calculated as:

Equation 1

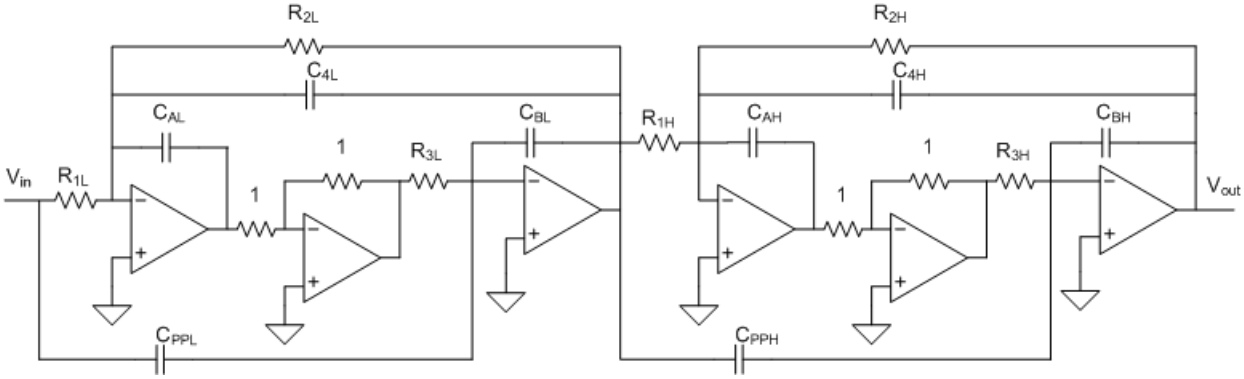
$$\frac{V_{OUT}}{V_{IN}} = \frac{s^2 + \omega_Z^2}{s^2 + \frac{s\omega_0}{Q}\omega_0^2}$$

Note Depending on whether Lower or Higher filter parts are calculated, L or H indexes can be used in all formulas instead of X.

In Equation 1, d is the damping ratio, ω0 is the natural frequency, and ωn is the normalized -3 dB frequency. All four-pole filters have out-of-band attenuation asymptotic to 24 dB for each octave (-12 dB per octave per pole). In-band performance is determined by damping ratio and natural frequency. The standard Butterworth filter has monotonic amplitude performance and maximally flat phase shift in the pass band. Filters with low damping ratios (Chebyshev) have flatter in-band amplitude characteristics, but non-linear phase shift in the pass band and pulse response characterized by ringing. Filters with high damping ratios (Bessel) have linear phase shift in the pass band and pulse response characteristics with minimum over-shoot, but reduced near out-of-band attenuation. Values for d and ω0 are readily available in any filter design reference. All of these filter forms can be realized by adjusting the ratios of capacitors in the switched capacitor PSoC blocks.

The basic form of the biquad filter is a pair of integrators with controlled DC and frequency-dependent feedback paths. The biquad can be understood by examining the standard RC form. There are two biquad blocks cascaded to create the four pole low-pass-filter on the following figure:

Figure 2. RC-Biquad Block Diagram



The typical RC-biquad low-pass uses three opamps. The transfer function of the double RC biquad is:

Equation 2

$$\frac{V_{OUT}}{V_{IN}} = \frac{-\frac{R_2}{R_1}\frac{1}{R_2 R_3 C_A C_B}}{s^2 + s\frac{C_4}{R_3 C_A C_B} + \frac{1}{R_2 R_3 C_A C_B}}$$

The typical RC-biquad low-pass uses three opamps. The transfer function of the double RC biquad is:

Equation 3

$$\frac{V_{OUT}}{V_{IN}} = \frac{V_{OUT_L}}{V_{IN_L}} \times \frac{V_{OUT_H}}{V_{IN_H}}$$

Note Depending on whether Lower or Higher filter parts are calculated, L or H indexes can be used in all formulas instead of X.

In the PSoC switched capacitor implementation, the center inverting opamp is eliminated by reversing the polarity of the gain of the output block. Resistors are transformed into the switched capacitors, as seen by comparing the schematics of the RC biquad and the switched capacitor implementation.

Because of the nature of switched capacitor circuits as time-sampled devices, the transfer function is developed in the time domain, where z^{-1} is the time delay of one sample period, rather than the frequency domain ($s=j\omega$). The transfer function is converted to the frequency domain using the bilinear transform. The transfer function resolves to the following.

Equation 4

$$\frac{V_{OUT_X}}{V_{IN_X}} = \frac{-\frac{C_{1_X}}{C_{2_X}} \frac{\left(1 + \left(\frac{s}{2f_s}\right)^2 \left(\frac{C_{PP_X} C_{A_X}}{C_{1_X} C_{3_X}} - \frac{1}{4}\right)\right) f_s^{\,2}}{\left(\frac{C_{A_X} C_{B_X}}{C_{2_X} C_{3_X}} - \frac{1}{4} - \frac{1}{2}\frac{C_{4_X}}{C_{2_X}}\right)}}{s^2 + \frac{C_{4_X}}{C_{2_X}} \frac{sf_s}{\left(\frac{C_{A_X} C_{B_X}}{C_{2_X} C_{3_X}} - \frac{1}{4} - \frac{1}{2}\frac{C_{4_X}}{C_{2_X}}\right)} + \frac{f_s^{\,2}}{\left(\frac{C_{A_X} C_{B_X}}{C_{2_X} C_{3_X}} - \frac{1}{4} - \frac{1}{2}\frac{C_{4_X}}{C_{2_X}}\right)}}$$

Evaluating this equation with the standard form of Equation 1 yields a set of design equations for gain, G, corner frequency $\omega n \omega 0$ and damping ratio, d.

Equation 5

$$G_X = -\frac{C_{1_X}}{C_{2_X}}$$

Equation 6

$$\omega_{n_X} \omega_0 = \frac{f_s (C_{2_X} C_{3_X})^{\frac{1}{2}}}{\left(C_{A_X} C_{B_X} - \frac{C_{2_X} C_{3_X}}{4} - \frac{C_{4_X} C_{3_X}}{2}\right)^{\frac{1}{2}}}$$

Equation 7

$$d_X = \frac{C_{4_X} \left(\frac{C_{3_X}}{C_{2_X}}\right)^{\frac{1}{2}}}{\left(C_{A_X} C_{B_X} - \frac{C_{2_X} C_{3_X}}{4} - \frac{C_{4_X} C_{3_X}}{2}\right)^{\frac{1}{2}}}$$

The numerator of Equation 3 has the term $1-(s/(2f_s))^2$, which results in reduced filter attenuation as the signal frequency approaches half of the Nyquist rate (that is, the sampling rate fS). Higher sample rates result in filter performance closer to the standard form of Equation 1 and smoother waveforms.

The elliptical zero for each section is calculated from the input data and the other already calculated values:

Equation 8

$$C_{PP_x} = \mathrm{int}\left(C_{1_x} C_{2_x} C_{A_x}\left(\left(\frac{f_{sample}}{2\pi f_{zero}} + 0.25\right)^2 + 0.5\right)\right)$$

In the previous equation, CPP_x, C1x, C2x, and CAx are capacitors values. For more information, see the ELPF4 block diagram. Fsample - Sample frequency value. Fzero - Frequency of zero point.

So actual location of the notch at the zero frequency is a function of capacitor ratios. The depth of the notch is limited only by parasitic capacitances in the SC blocks. The notch adds a frequency dependent term in the numerator of the transfer function:

Equation 9

$$H(S) = 1 - \left(\frac{f_{signal}}{f_{zero}}\right)^2$$

The closer the zero is to the nominal corner frequency, the greater the attenuation. If absolute band flatness is required, the user can compensate this effect by changing the damping factor of the transfer function poles by using the custom filter configuration.

Power Setting for Applications

Switched capacitor block power and bias settings determine the performance of the filters. The power mode is selected using user module APIs. The bias is selected using the "Opamp Bias" parameter in the global resources window of PSoC Designer. Power and bias settings determine the opamp operating current, which in turn determines the Slew Rate (SR). The opamp slew rate is 4 V/µsec when power and bias are both set High. It is reduced by half for each step down in power and bias.

The slew rate required to faithfully deliver the signal is dependent on peak voltage (V_{pk}) and frequency (F). The filter's output signal slews from its old value to its new value during one half of one phase of the clock. As a result, the filter requires an opamp slew rate four times the slew rate of the signal.

$$SR = 4 \times (2\pi V_{pk} F)$$

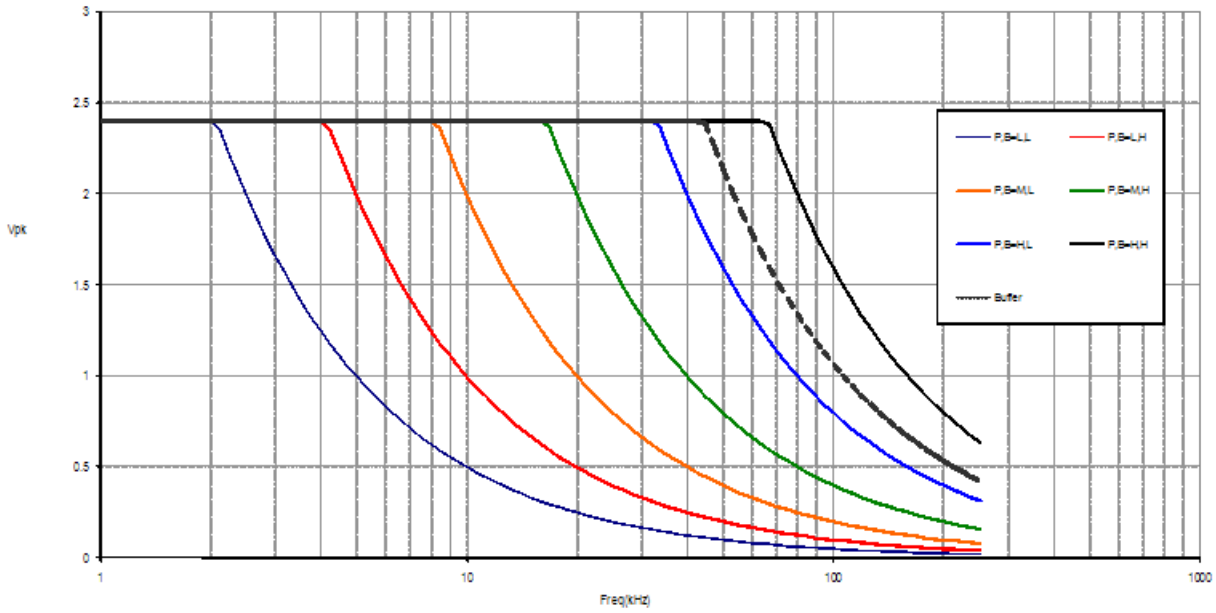
The maximum operating frequency for rail-rail output for V_{DD} = 5.0 V is given in the following table:

Power,Bias	kHz
P,B=H,H	64
P,B=H,L	32

Power,Bias	kHz
P,B=M,H	16
P,B=M,L	8
P,B=L,H	4
P,B=L,L	2

In this table, P = Power, B = Opamp Bias, L = Low, H = High and M = Medium. For lower signal levels, the allowed peak voltage is shown in the following graph.

Figure 3. Peak Voltage for Lower Signal Levels



The filter output is an internal signal. It must be buffered if driven off-chip. The buffer has a slew rate of 0.65 V/µsec. The operating frequency limits of the buffer are shown in the previous graph.

Filter Design

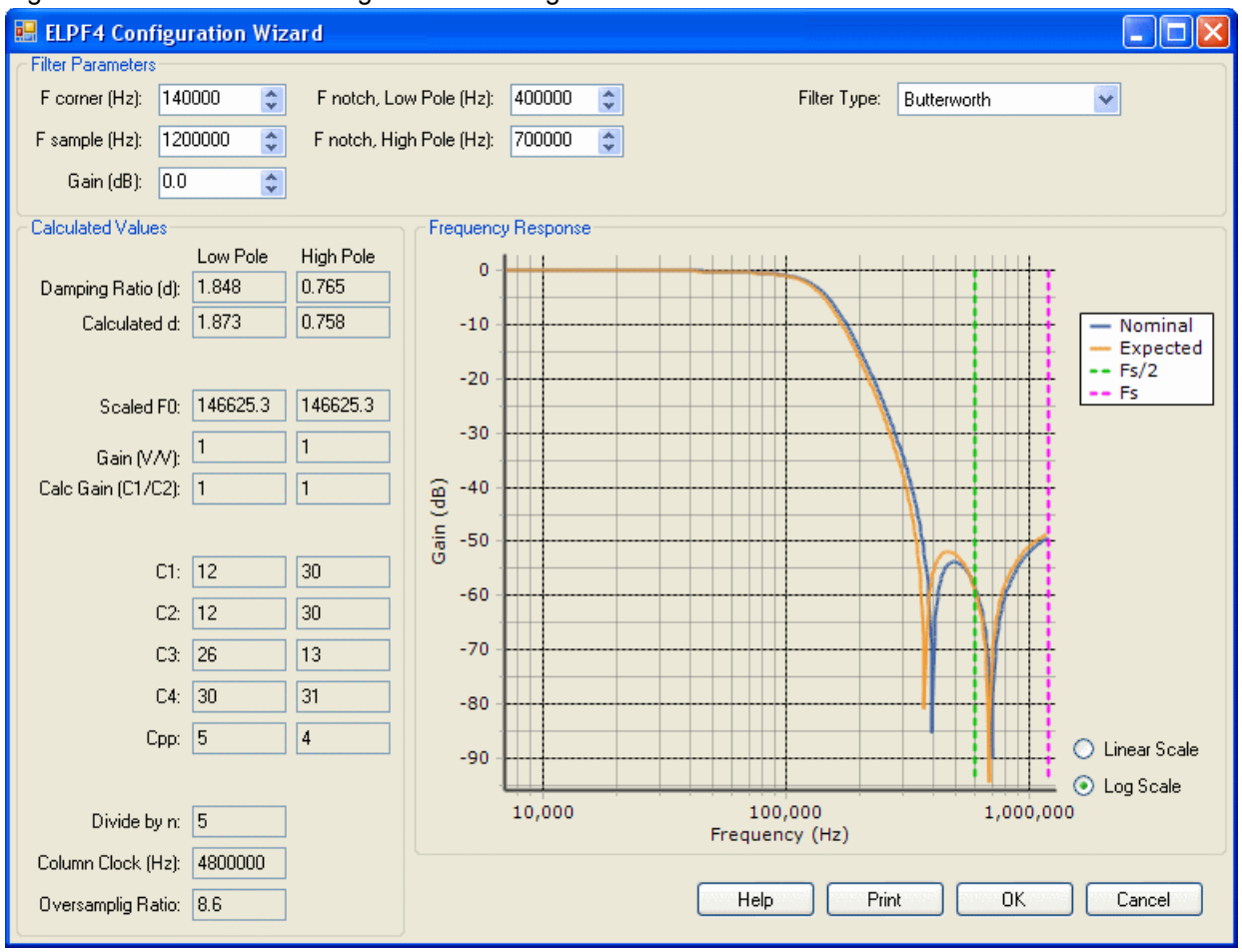
The design objective is usually to achieve the highest possible sample rate (f_{clk}), for the best waveform fidelity and minimum aliasing of out-of-band signals. Other system requirements (for example, shared clocks) may determine sample rates; capacitor values may be tailored to achieve the required sample rate.

The ELPF4 provides three alternatives for determining the capacitor values. PSoC Designer provides a filter design wizard to automate the procedure for four-pole filters. This same procedure is implemented in the spreadsheet, *ELPF4 Design.xls*, which may be obtained from the “Documentation...” entry in PSoC Designer’s Help menu. Design constraints enforced by the wizard may be modified experimentally in the spreadsheets. For the ultimate in hands-on control over the design process, see the appendix at the end of this document for a numerical procedure that may be carried out manually. It also provides an example showing how the procedure works for a Butterworth filter with a 1 kHz corner frequency.

To use the PSoC Designer’s built-in Filter Design Wizard, first place an ELPF4 instance in the analog array. Right-click on the user module and choose “ELPF4 Wizard...” from the pop up menu. The resulting

dialog, shown in the following figure, describes a simple iterative procedure for designing the transfer function.

Figure 4. ELPF4 Filter Design Wizard Dialog Box



Scrolling down in the dialog reveals the table of values used to plot the magnitude response. Values from this table may be cut and pasted into spreadsheets or other tools for further graphing and analysis.

DC and AC Electrical Characteristics

The following values are indicative of expected performance and based on initial characterization data. Unless otherwise specified, all limits guaranteed for TJ =+25 °C, VDD = 5.0 V, Power HIGH, Opamp bias LOW, output referenced to Analog Ground = 2*VBandGap.

Table 1. 5.0 V ELPF4 DC Electrical Characteristics

Parameter	Typical	Limit	Units	Conditions and Notes
DC Offset Voltage[1]	18	--	me	Reference to Analog Ground
DC Gain Error[2]	1.55	--	%	

Parameter	Typical	Limit	Units	Conditions and Notes
Operating Current				
Low Power	290	--	µA	
Medium Power	1065	--	µA	
High Power	4015	--	µA	

Table 2. 5.0 V ELPF4 AC Electrical Characteristics

Parameter	Typical	Limit	Units	Conditions and Notes
Maximum Clock Frequency[3]				
Low Power	--	2	MHz	
Medium Power	--	4	MHz	
High Power	--	8	MHz	
Corner Frequency Error	2.3	--	%	Deviation from Nominal[2]
Damping Ratio Error	1.85	--	%	
Noise[4]	2140	--	nV/√Hz	

The following values are indicative of expected performance and based on initial characterization data. Unless otherwise specified in the following tables, limits guaranteed for TA = 25 °C, VDD = 3.3 V, Power HIGH, Opamp bias LOW, output referenced to Analog Ground = VDD/2.

Table 3. 3.3 V ELPF4 DC Electrical Characteristics

Parameter	Typical	Limit	Units	Conditions and Notes
DC Offset Voltage[1]	14	--	mV	Reference to Analog Ground
DC Gain Error[2]	1.65	--	%	
Operating Current				
Low Power	225	--	µA	
Medium Power	850	--	µA	
High Power	3275	--	µA	

Table 4. 3.3 V ELPF4 AC Electrical Characteristics,

Parameter	Typical	Limit	Units	Conditions and Notes
Maximum Clock Frequency[3]				
Low Power	--	2	MHz	
Medium Power	--	4	MHz	
High Power	--	8	MHz	
Corner Frequency Error	2.15	--	%	Deviation from Nominal[2]
Damping Ratio Error	1.9	--	%	
Noise[4]	1880	--	nV/√Hz	

Note Electrical Characteristics Notes

1. Deviation values determined from nominal filter: fcenter=1 kHz Butterworth, unity gain, C1=1, C2=3, C3=31, C4=1, fclock=20.3 kHz, Q=10.
2. Sample rate is one fourth of column clock frequency.
3. Noise found at 1 kHz using a 10 kHz filter.

Unless otherwise specified, all limits guaranteed for T_J = -40 °C to, +85 °C, V_{DD} = 5.0 V ±10%, Power HIGH, Opamp bias LOW, output referenced to Analog Ground = 2*$V_{BandGap}$.

Table 5. 5.0 V ELPF4 DC Electrical Characteristics

Parameter	Typical[1]	Limit[2]	Units	Conditions and Notes
Offset Voltage[5]	42		mV	See graph
DC Gain Error	1.55		%	Ref Design = 1.0 kHz Butterworth[3]
Operating Current				
Low Power	250	--	µA	
Medium Power	560	--	µA	
High Power	1560	2000	µA	

Table 6. 5.0 V ELPF4 AC Electrical Characteristics

Parameter	Typical[1]	Limit[2]	Units	Conditions and Notes
Corner Frequency Deviation	--	2.5	%	Deviation from Nominal
Damping Ratio Deviation	--	2.6	%	Deviation from Nominal
Noise	--	8.0	mV rms	

Parameter	Typical[1]	Limit[2]	Units	Conditions and Notes
Clock Frequency				
Low Power	--	2.0	MHz	
Medium Power	--	4.0	MHz	
High Power	--	8.0	MHz	

Unless otherwise specified in the following tables, limits guaranteed for TA = -40 °C to +85 °C, V_{DD} = 3.0 to 3.6 V, Power HIGH, Opamp bias LOW, output referenced to Analog Ground = $V_{DD}/2$.

Table 7. 3.3 V ELPF4 DC Electrical Characteristics

Parameter	Typical[1]	Limit[2]	Units	Conditions and Notes
Offset Voltage[5]	42		mV	See graph
DC Gain Error	1.65		%	Ref Design = 1.0 kHz Butterworth[3]
Operating Current				
Low Power	200	--	µA	
Medium Power	500	--	µA	
High Power	1280	1800	µA	

Table 8. 3.3 V ELPF4 AC Electrical Characteristics

Parameter	Typical[1]	Limit[2]	Units	Conditions and Notes
Corner Frequency Deviation[3]	--	2.5	%	
Damping Ratio Deviation[3]	--	2.6	%	
Noise	--	6.0	mV rms	
Clock Frequency[4]				
Low Power	--	1.0	MHz	
Medium Power	--	2.0	MHz	
High Power	--	4.0	MHz	

Note Electrical Characteristics

1. Typical values represent parametric norm at +25 °C.
2. Limits are guaranteed by testing or statistical analysis.
3. Reference design C1=1, C2=1, C3=2, C4=31, CA=32, CB=32, f(sample)=139.2 kHz, damping ratio = 1.392.

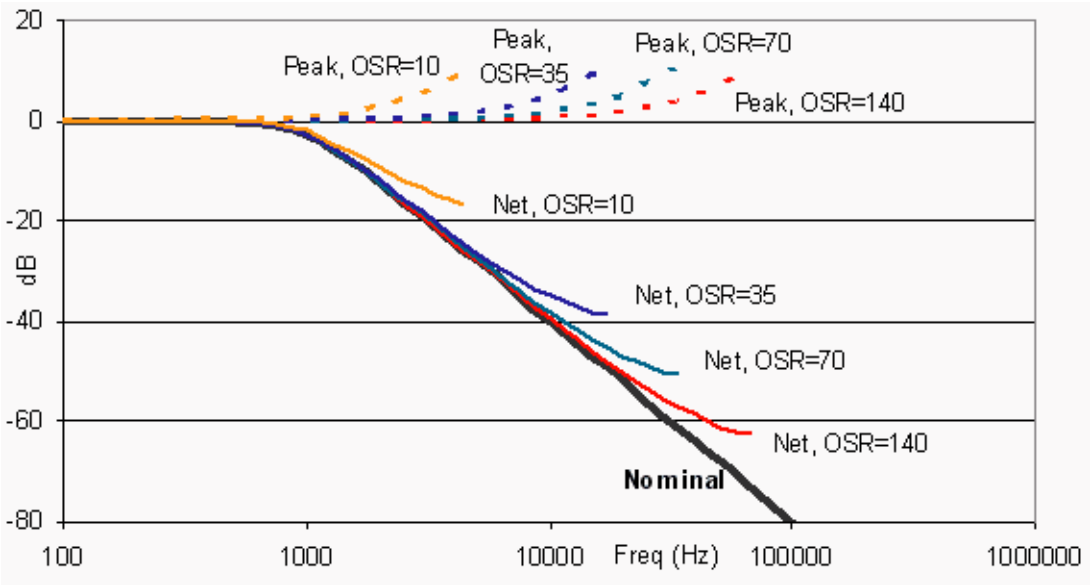
4. Clock frequency is for filter of Note 3, with corner frequency scaled with clock.
5. 1 kHz filter, fclk=35 kHz, horizontal inverting topology, C1=C2=8, C3=16, C4=31, CA=CB=32.

Performance Notes

Errors in gain and phase for the filter are only significant near the corner frequency. Other selections can be made for filter capacitor values. C2x and C3x ratios can be adjusted to allow better resolution in the damping ratio. Setting CAx or CBx equal to 16 allows for additional resolution in the damping ratio setting, but at the cost of reducing sample rate from the maximum possible value.

The gains in signal fidelity, as a result of increased sample rate, may out-weigh transfer function error as a result of non-ideal pole location. Lowering the sampling rate (by increasing C1x and C2x) has the effect of reducing the attenuation approaching the Nyquist rate.

Figure 5. Filter Performance versus Oversampling Ratio



Filter performance deviates by some degree from the standard form as a result of the effects of opamp open loop gain, absolute capacitor value and filter placement. While low-pass filters above 100 kHz are mathematically possible, opamp performance limitations make these choices unstable. Filters with corner frequencies above 40 kHz should have user module power set to “HIGHPOWER” and Opamp Bias set to “High” in the Global Parameters window. For a given filter characteristic there may be a number of capacitor value solutions which will meet the requirement, but each variation will have a different sample rate. In general, increasing the capacitor values decreases the oversampling ratio (that is, ratio of sample rate to nominal -3dB frequency). For the highest oversample ratio, and smoothest output waveform, the filter should be built starting with the lowest possible value of C2x in either the design spreadsheet, design wizard or "by hand" derivation using the design equations. For example, a Butterworth filter with C2x=1 and C4x=31 has an oversample ratio of 140. Setting C2x=2 results in oversample ratio of 70. In all cases, the selected sample rate must be less than the maximum of 1.00 MHz (column clock = 4.0 MHz); higher frequency filters require lower oversample ratios to meet this requirement. The numerator of Equation 3 shows a pair of complex zeros at 2fs. As the sample frequency is reduced, the effect of these zeros becomes ever stronger, peaking the upper band edge as shown in the following figure.

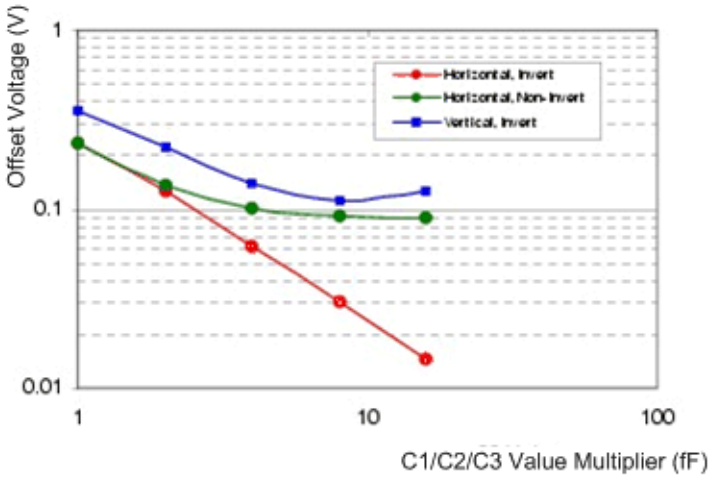
Charge injection in the internal switch topology limits DC performance of the filter. This effect has a random component and a placement determined component.

Note Filters placed in horizontal block pairs have lower DC offset than filters placed in vertical block pairs. Scaling the filter by increasing the value of C2x (and C1x and C3x), results in reduced DC

offset error. See the following figure for graphical representation of how the value of C2 affects the offset error.

The C1, C2, and C3 values are multipliers which are entered into the FPF2 wizard. These multipliers are coefficients with Csc from the DC Analog PSoC Block Specifications from the device datasheet.

Figure 6. Typical Offset Dependence on Capacitor Value (C1/C2/C3) for PSoC Devices



Placement

The Device Editor maps the logical FLIN and FLOUT blocks onto a pair of adjacent switched capacitor PSoC blocks in the device's analog array. ELPF4 UM uses two pairs of the FLIN and FLOUT blocks named FLINL, FLOUTL for first pair and FLINH, FLOUTH for second. So ELPF4 takes four SC blocks for one instance. There are several ways to construct the four-pole elliptical low-pass filter circuit out of the analog PSoC blocks. Each construction uses different capacitors and connections within the FLIN and FLOUT blocks. Each results in a different circuit topology with different mapping and I/O consequences. The most noticeable difference is whether the two PSoC blocks lie in a row or column of the analog array. The topologies also determine which connections can be made to other blocks in the array. Regardless of the selected placement, however, the filter inputs always connect to the FLINL block and outputs are driven by the FLOUTH block.

Wizard Parameters and Characteristics

Filter Input Parameters and Characteristics

Figure 7. Standard Filter Approximation

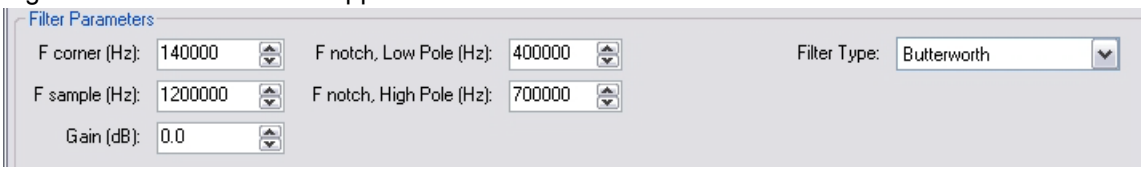


Figure 8. Custom Complex Pole

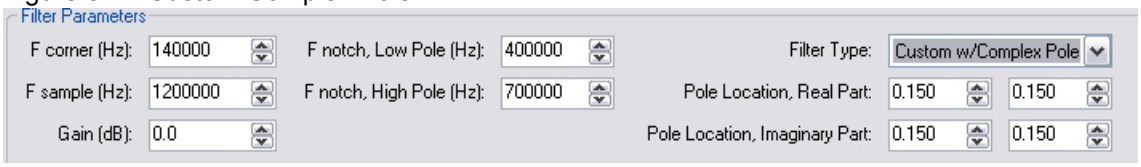


Figure 9. Custom Real Pole



F corner

Enter the desired corner frequency for your filter.

Notch Freq of Low Pole (Hz)

Enter the desired notch frequency for your filter's low pole.

Notch Freq of High Pole (Hz)

Enter the desired notch frequency for your filter's high pole.

F sample

Enter the desired sample frequency for your filter.

Gain (dB)

Enter the desired gain for your filter in dB.

Filter Type

Select the type of approximation you want to use in your design.

Real Part (Pole Location of)

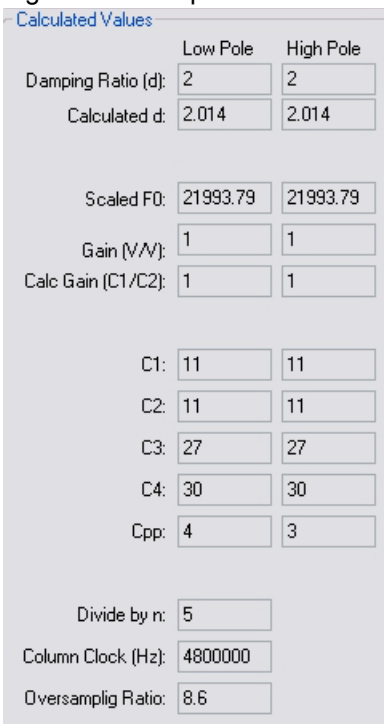
Enter the real part of the filter pole location on a complex plane when customizing filter characteristics.

Imaginary Part (Pole Location of)

Enter the imaginary part of the filter pole location on a complex plane when customizing the filter characteristics.

Filter Output Parameters and Characteristics

Figure 10. Output Parameters



Damping Ratio (d)

Nominal damping ratio of the filter for lower and higher pairs of SC blocks respectively.

Calculated d

Expected damping ratio of the designed filter for lower and higher pairs of SC blocks respectively.

Scaled F0

Expected roll-of frequency of the designed filter for lower and higher pairs of SC blocks respectively.

Gain(V/V)

Requested voltage gain of the filter for lower and higher pairs of SC blocks respectively.

Calc Gain (C1/C2)

Expected voltage gain of the designed filter for lower and higher pairs of SC blocks respectively.

C1

C1 capacitance value of designed filter: The value in the column marked “Low Pole” is transferred to the C1L user module parameter. The value in the column marked “High Pole” is transferred to the C1H user module parameter.

C2

C2 capacitance value of designed filter: The value in the column marked “Low Pole” is transferred to the C2L user module parameter. The value in the column marked “High Pole” is transferred to the C2H user module parameter.

C3

C3 capacitance value of designed filter: The value in the column marked “Low Pole” is transferred to the C3L user module parameter. The value in the column marked “High Pole” is transferred to the C3H user module parameter.

C4

C4 capacitance value of designed filter: The value in the column marked “Low Pole” is transferred to the C4L user module parameter. The value in the column marked “High Pole” is transferred to the C4H user module parameter.

CPP

CPP capacitance value of designed filter: The value in the column marked “Low Pole” is transferred to the CPPL user module parameter. The value in the column marked “High Pole” is transferred to the CPPH user module parameter.

Divide by n

Calculated divider for the filter input clock. Use this divider number when configuring PSoC clock dividers and the column clock where the filter is placed.

Column Clock (Hz)

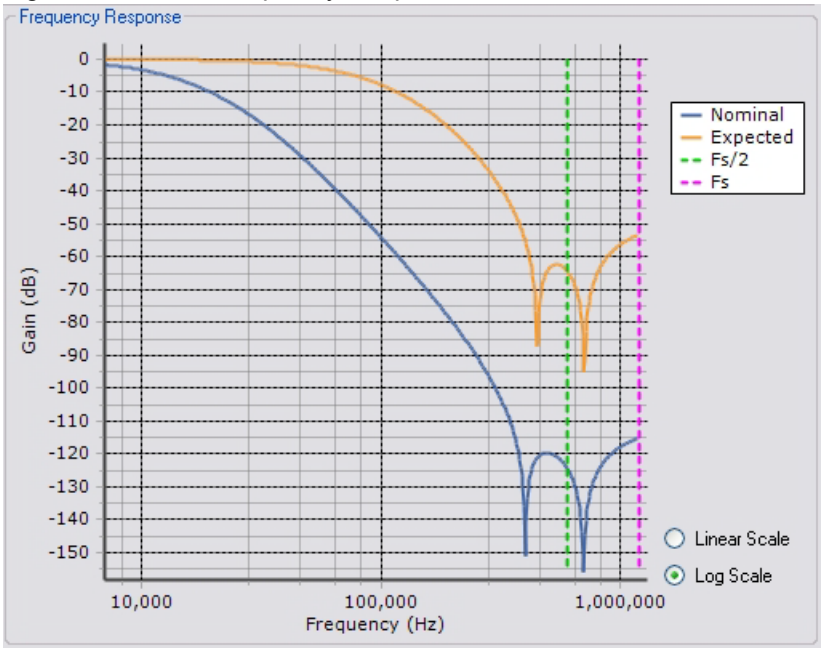
Calculated column clock for the designed filter. Use this value when configuring PSoC clock dividers and the column clock where the filter is placed.

Oversampling Ratio

The oversampling ratio of the designed switched-capacitor filter.

Filter Response Plot

Figure 11. Filter Frequency Response Plot



Frequency response of designed filter

The area where the two frequency response plots are displayed – one for the nominal (desired) filter and one for expected (capable to be implemented on PSoC).

Frequency scale selector

Select the frequency axis scale, which is most representative for you.

Wizard Controls

Figure 12. Configuration Wizard Controls Panel



Help

This button calls this user module datasheet, which has information on how to use the user module wizard. The datasheet's Wizard section describes wizard usage.

OK

This button applies the parameters of the designed filter to the ELPF4 User Module if the obtained characteristics meet the requirements of your design.

Print

Press this button if you need to print the Filter Configuration window with all parameters and plot data.

Cancel

This button closes the Filter Configuration window without any modifications of ELPF4 User Module.

Parameters and Resources

To make a four-pole low-pass filter, place an instance of the ELPF4 User Module in the Device Editor's analog array. Use one of the design procedure options to determine the values for the filter's capacitors, then connect the inputs and configure the analog bus connection and clock resources. Each of these parameters are discussed in this section.

Input

Inputs to the filter are driven by the outputs of the adjacent PSoC blocks. You can make input selections in the Device Editor.

AnalogBus

The output of the FLOUTH block can be connected to the analog column output bus. This enables connection to the Analog Output Buffer for the same column and prevents analog output-bus access of other user modules in the same column. All interconnections are configured using the Device Editor.

Capacitor Values C1L, C2L, C3L, C4L, CPPL, C1H, C2H, C3H, C4H, and CPPH

The ratios of these ten capacitor values determine the frequency and phase response of the filter. The names refer to the capacitors drawn in the ELPF4 Schematic Diagram. C1L through C4H including CPPL and CPPH take values from 0 to 31 (though values greater than zero are required for meaningful transfer functions). Design of the transfer function may be accomplished using automated or manual procedures. To access the built in design tool, right-click on the placed filter and choose "ELPF4 Wizard..." from the pop up menu. See the Filter Design section for more information on design.

Sample Frequency (column clock/4)

The required Sample Frequency, equivalent to the column clock divided by 4, for the filter is calculated using the design equations in the Functional Description section. Unlike the other user module param-

eters, the Sample Frequency does not appear in the list of user module parameters underneath the Device Editors list of Global Resources. In addition, unlike signal inputs that are specific to a particular user module, the Sample Frequency (column clock/4) serves an entire analog column. The column clocks for all filter PSoC blocks must be the same.

Where horizontal placement of blocks is selected, both column clocks must be driven from the same source to have the same sample frequency. Each column-clock generator divides its input by four to produce φ1 and φ2, the internal clocks in the blocks, so the source must be four times faster than the desired filter sample frequency.

Choices for the clock source include any of the digital PSoC blocks and the system clock dividers. All of the Timer, Counter, and Pulse-Width Modulator (PWM) User Modules are suitable choices when system clock dividers must be consigned to other uses.

The clock source to the column clock is selected using the CLK multiplexer, for each column in the Device Editor. The system clocks are direct inputs to this multiplexer. When PSoC blocks are used for clock generation, they are connected through the ACLK0 and ACLK1 multiplexers to the CLK multiplexer.

Gain

This property is available from the GUI only. It provides gain setting.

Corner Frequency

This property is available from the GUI only. It provides filter corner frequency selection in range from 20 Hz to 150 kHz.

Notch Frequency for Low and High pole.

This property is available from the GUI only. It provides filter notch frequency selection in range from 20 Hz to 600 kHz.

Filter type

This property is available from the GUI only. It provides selection of classical all-pole filter configurations (Butterworth, Bessel, and Chebyshev). You can also select custom filter configuration and enter values for real or complex poles.

Application Programming Interface

The Application Programming Interface (API) routines are provided as part of the user module to allow the designer to deal with the module at a higher level. This section specifies the interface to each function together with related constants provided by the “include” files.

Note

In this, as in all user module APIs, the values of the A and X register may be altered by calling an API function. It is the responsibility of the calling function to preserve the values of A and X prior to the call if those values are required after the call. This “registers are volatile” policy was selected for efficiency reasons and has been in force since version 1.0 of PSoC Designer. The C compiler automatically takes care of this requirement. Assembly language programmers must ensure their code observes the policy, too. Though some user module API function may leave A and X unchanged, there is no guarantee they may do so in the future.

For Large Memory Model devices, it is also the caller's responsibility to preserve any value in the CUR_PP, IDX_PP, MVR_PP, and MVW_PP registers. Even though some of these registers may not be modified now, there is no guarantee that will remain the case in future releases.

Entry points are provided to initialize the ELPF4 User Module, change power settings, and disable the user module.

ELPF4_Start

Description:

Performs all required initialization for this user module and sets the power level for the switched capacitor PSoC blocks.

C Prototype:

```
void  ELPF4_Start(BYTE bPowerSetting)
```

Assembly:

```
mov   A, bPowerSetting
lcall  ELPF4_Start
```

Parameters:

bPowerSetting: One byte that specifies the power level to both analog PSoC blocks. Following reset and configuration, the PSoC blocks assigned to the instrumentation amplifier are powered down. Symbolic names provided in C and assembly, and their associated values, are listed in the following table.

Symbolic Name	Value
ELPF4_LOWPOWER	1
ELPF4_MEDPOWER	2
ELPF4_HIGHPOWER	3

Note For proper performance, filters with corner frequencies above 40 kHz should use ELPF4_HIGHPOWER and set the global parameter “Opamp Bias” to High in the Global Parameters window.

Return Value:

None

Side Effects:

You can alter the A and X registers by this function.

ELPF4_SetPower

Description:

Sets the power level for the switched capacitor PSoC blocks. May be used to turn the blocks in the user module off and on.

C Prototype:

```
void  ELPF4_SetPower(BYTE bPowerSetting)
```

Assembly:

```
mov   A, bPowerSetting
lcall  ELPF4_SetPower
```

Parameters:

bPowerSetting: One byte that specifies the power level to both analog PSoC blocks.

Return Value:

None

Side Effects:

You can alter the A and X registers by this function.

ELPF4_SetC1L, SetC2L, SetC3L, SetC4L, SetCPPL, SetC1H, SetC2H, SetC3H, SetC4H and SetCPPH

Description:

Sets the value of specific capacitors in the user module. This allows adjustment of gain by modifying C1, and alteration of filter transfer characteristics by adjusting the other values.

C Prototype:

```
void  ELPF4_SetC1L(BYTE bCapValue)
void  ELPF4_SetC2L(BYTE bCapValue)
void  ELPF4_SetC3L(BYTE bCapValue)
void  ELPF4_SetC4L(BYTE bCapValue)
void  ELPF4_SetCPPL(BYTE bCapValue)
void  ELPF4_SetC1H(BYTE bCapValue)
void  ELPF4_SetC2H(BYTE bCapValue)
void  ELPF4_SetC3H(BYTE bCapValue)
void  ELPF4_SetC4H(BYTE bCapValue)
void  ELPF4_SetCPPH(BYTE bCapValue)
```

Assembly:

```
mov   A, bCapValue
lcall  ELPF4_SetC1L               ; or, lcall  ELPF4_SetC2L (or SetC3L or SetC4L or
SetCPPL etc.)
```

Parameters:

bCapValue: Integer value from 0 to 31 for C1L, C2L, C3L, C4L, CPPL, C1H, C2H, C3H, C4H and CPPH (see the ELPF4 Schematic Drawing). Values outside this range will be truncated to 31.

Return Values:

None

Side Effects:

You can alter thee A and X registers by this function.

ELPF4_Stop

Description:

Powers the user module off.

C Prototype:

```
void  ELPF4_Stop(void)
```

Assembly:

```
lcall  ELPF4_Stop
```

Parameters:

None

Return Value:

None

Side Effects:

You can alter the A and X registers by this function.

Sample Firmware Source Code

In C, using the low-pass filter is as simple as using the Start API to begin operation and calling the Stop API when done.

```
//
//  This sample shows how to create a Low Pass Filter with corner frequency 1kHz.
//
// OVERVIEW:
//  The ELPF4 input/output can be routed to any analog pin or adjacent analog block
// depending on placement.
//  In this example the ELPF4 input is routed to P0[5] and the output is routed to P0[3].
//
//The following changes need to be made to the default settings in the Device Editor:
//
//   1. Place the ELPF4 user module onto ASC10, ASD11, ASD20 and ASC21 blocks.
//   2. Rename the User Module's instance name to ELPF4.
//   3. Run the ELPF4 Wizard from the context menu.
//      - Set the F corner (Hz) parameter to 1000
//      - Set the F sample (Hz) parameter to 100000
//      - Set the F notch, Low Pole (Hz) parameter to 2000
//      - Set the F notch, High Pole (Hz) parameter to 4000
//      - Leave the Gain and Filter Type parameters by default
//  ("0.0" and "Butterworth" respectively)
//      - Click the "OK" button
//   4. Set the ELPF4's Input parameter to ACB00.
//   5. Set the ELPF4's AnalogBus parameter to AnalogOutBus_0.
//   6. Leave the rest of UM parameters by default.
//   7. Set the AnalogColumn_Clock_0 to VC2 (on the interconnect view)
//   8. Set the AnalogOutBuf_0 to Port_0_3 (on the interconnect view).
//   9. Place the PGA UM onto ACB00 block.
```

```
//   10. Rename the User Module's instance name to PGA.
//   11. Set the PGA's Gain parameter to 1.000.
//   12. Set the PGA's Input parameter to AnalogColumn_InputMUX_0.
//   13. Set the PGA's Reference parameter to AGND.
//   14. Set the PGA's AnalogBus parameter to Disable.
//   15. Set the AnalogColumn_InputMux_0 to Port_0_5 (on the interconnect view)
//
// CONFIGURATION DETAILS:
//
//   1. The UM's instance names have to be shortened to ELPF4 and PGA.
//   2. The Analog Column clock should be 400 kHz to get the 1kHz low Pass Filter with
//   Over Sampling Ration = 100.
//
// PROJECT SETTINGS:
//
//   1. Set the SysClock Source to Internal 24_MHz       System clock is set to 24MHz
//   2. Set the VC1=SysClk/N to 16 (default)
//   3. Set the VC2=VC1/N to 15
//
//
// USER MODULE PARAMETER SETTINGS:
//
//  ------------------------------------------------------------------------------
//  UM         Parameter         Value                 Comments
//  ------------------------------------------------------------------------------
//  ELPF4      Name              ELPF4                 UM's instance name
//             Input             ACB00
//             C1L               1                     Set by Wizard
//             C2L               1                     Set by Wizard
//             C3L               4                     Set by Wizard
//             C4L               29                    Set by Wizard
//             CPPL              8                     Set by Wizard
//             C1H               2                     Set by Wizard
//             C2H               2                     Set by Wizard
//             C3H               2                     Set by Wizard
//             C4H               24                    Set by Wizard
//             CPPH              2                     Set by Wizard
//             AnalogBus         AnalogOutBus_0
//
// PGA         Name              PGA                   UM's instance name
//             Gain              1.000
//             Input             AnalogColumn_InputMUX_0
//             Reference         AGND
//             AnalogBus         Disable
//  ------------------------------------------------------------------------------

/* Code begins here */

#include <m8c.h>        // part specific constants and macros
#include "PSoCAPI.h"    // PSoC API definitions for all User Modules

void main(void)
{
    // M8C_EnableGInt ;           // Uncomment this line to enable Global Interrupts
    PGA_Start(PGA_HIGHPOWER);     // Turn on the PGA UM
```

```
    ELPF4_Start(ELPF4_HIGHPOWER);           // Turn on the ELPF2 UM
}
```

The equivalent assembly language code is:

```
;
; DESCRIPTION:
;
;  This sample shows how to create a Low Pass Filter with corner frequency 1kHz.
;
; OVERVIEW:
;
;  The ELPF4 input/output can be routed to any analog pin or adjacent analog block
;  depending on placement.
;  In this example the ELPF4 input is routed to P0[5] and the output is routed to P0[3].
;
;The following changes need to be made to the default settings in the Device Editor:
;
;   1. Place the ELPF4 user module onto ASC10, ASD11, ASD20 and ASC21 blocks.
;   2. Rename the User Module's instance name to ELPF4.
;   3. Run the ELPF4 Wizard from the context menu.
;      - Set the F corner (Hz) parameter to 1000
;      - Set the F sample (Hz) parameter to 100000
;      - Set the F notch, Low Pole (Hz) parameter to 2000
;      - Set the F notch, High Pole (Hz) parameter to 4000
      - Leave the Gain and Filter Type parameters by default
;      ("0.0" and "Butterworth" respectively)
;      - Click the "OK" button
;   4. Set the ELPF4's Input parameter to ACB00.
;   5. Set the ELPF4's AnalogBus parameter to AnalogOutBus_0.
;   6. Leave the rest of UM parameters by default.
;   7. Set the AnalogColumn_Clock_0 to VC2 (on the interconnect view)
;   8. Set the AnalogOutBuf_0 to Port_0_3 (on the interconnect view).
;   9. Place the PGA UM onto ACB00 block.
;   10. Rename the User Module's instance name to PGA.
;   11. Set the PGA's Gain parameter to 1.000.
;   12. Set the PGA's Input parameter to AnalogColumn_InputMUX_0.
;   13. Set the PGA's Reference parameter to AGND.
;   14. Set the PGA's AnalogBus parameter to Disable.
;   15. Set the AnalogColumn_InputMux_0 to Port_0_5 (on the interconnect view)
;
; CONFIGURATION DETAILS:
;
;   1. The UM's instance names have to be shortened to ELPF4 and PGA.
;   2. The Analog Column clock should be 400 kHz to get the 1kHz low Pass Filter with
;   Over Sampling Ration = 100.
;
; PROJECT SETTINGS:
;
;   1. Set the SysClock Source to Internal 24_MHz       System clock is set to 24MHz
;   2. Set the VC1=SysClk/N to 16 (default)
;   3. Set the VC2=VC1/N to 15
;
;
; USER MODULE PARAMETER SETTINGS:
```

```
;
;  ----------------------------------------------------------------------------
;  UM         Parameter         Value                 Comments
;  ----------------------------------------------------------------------------
;  ELPF4      Name              ELPF4                 UM's instance name
;             Input             ACB00
;             C1L               1                     Set by Wizard
;             C2L               1                     Set by Wizard
;             C3L               4                     Set by Wizard
;             C4L               29                    Set by Wizard
;             CPPL              8                     Set by Wizard
;             C1H               2                     Set by Wizard
;             C2H               2                     Set by Wizard
;             C3H               2                     Set by Wizard
;             C4H               24                    Set by Wizard
;             CPPH              2                     Set by Wizard
;             AnalogBus         AnalogOutBus_0
;
;  PGA        Name              PGA                   UM's instance name
;             Gain              1.000
;             Input             AnalogColumn_InputMUX_0
;             Reference         AGND
;             AnalogBus         Disable
;  ----------------------------------------------------------------------------

; Code begins here

include "m8c.inc"       ; part specific constants and macros
include "memory.inc"    ; Constants & macros for SMM/LMM and Compiler
include "PSoCAPI.inc"   ; PSoC API definitions for all User Modules

export _main

_main:

    ; M8C_EnableGInt ; Uncomment this line to enable Global Interrupts
    mov   A, PGA_HIGHPOWER
    lcall PGA_Start
    mov   A, ELPF4_HIGHPOWER
    lcall ELPF4_Start

    ; Insert your main assembly code here.

.terminate:
    jmp .terminate
```

Note The design equations show that gain is proportional to the value of C1, but the corner frequency and damping do not depend on it. Once the transfer function is chosen, the ELPF4_SetC1L API function may be used to implement a programmable-gain control.

Configuration Registers

The topology and placement of the ELPF4 User Module determines over half the bits in the configuration registers for the analog switched capacitor PSoC blocks used. Of those, the ones that are independent of placement location are indicated by fixed values in the register tables. Of the variable bitfields, most are determined by selection of input and transfer function design. Definitions of the variable bitfields used in the register definitions follow, at the end of this section.

Table 9. Block FLINL: Register CR

Bit	7	6	5	4	3	2	1	0
CR0	CAL	0	Polarity	C1L				
CR1	Input			C2L				
CR2	0	0	0	0	0	0	0	0
CR3	0	0	1	0	Feedback		Power	

Table 10. Block FLOUTL: Register CR

Bit	7	6	5	4	3	2	1	0
CR0	CBL	0	0	C3L				
CR1	FBINL			CPPL				
CR2	0	0	0	C4L				
CR3	0	0	1	0	0	1	Power	

Table 11. Block FLINH: Register CR

Bit	7	6	5	4	3	2	1	0
CR0	CAH	0	0	C1H				
CR1	FLOUTL			C2H				
CR2	0	0	0	0	0	0	0	0
CR3	0	0	1	0	Feedback		Power	

Table 12. Block FLOUTH: Register CR

Bit	7	6	5	4	3	2	1	0
CR0	CBH	0	0	C3H				
CR1	FBINH			CPPH				
CR2	AnalogBus	CompBus	0	C4H				
CR3	0	0	1	0	0	1	Power	

Variable BitField Definitions

The following definitions apply to all preceding register definitions:

CAL, **CAH, CBL**, and **CBH** set the FLIN and FLOUT feedback capacitors, respectively, to either 16 or 32 units (see the ELPF4 Schematic Diagram).

C1L, **C2L**, **C3L**, **C4L, CPPL, C1H**, **C2H**, **C3H, C4H and CPPH** set the capacitors illustrated in the ELPF4 Schematic Diagram to integer values between 1 and 31. They are configured directly in the Device Editor or indirectly through use of the filter Design Wizard.

Input controls the multiplexor that selects the input signal to be conditioned by the ELPF4 User Module. The user module “Input” parameter determines the value of this bitfield. The value of the Input parameter is manually configured using the Device Editor.

AnalogBus enables connection of the filter output to the analog bus. The user module “AnalogBus” parameter determines the value of this bitfield. The value of the AnalogBus parameter is manually configured using the Device Editor.

CompBus enables connection of the filter output to the comparator bus. This property is not used in current user module.

Feedback is the C2L and C2H feedback connection, automatically determined by placement of the ELPF4 User Module in the Device Editor.

FBINL is the connection from the FLIN output to the FLOUT input, automatically determined by placement of the ELPF4 User Module in the Device Editor.

FLOUTL is the connection from the FLOUTL output to the FBINH input, automatically determined by placement of the ELPF4 User Module in the Device Editor.

FBINH is the connection from the FLINH output to the FLOUTH input, automatically determined by placement of the ELPF4 User Module in the Device Editor.

Polarity controls whether the output of the filter is inverted or not. This property is not used in current user module.

Power controls the On/Off state of the PSoC block and bias current setting. It is set initially by calling the user module API function ELPF4_Start and can be modified by calling the functions ELPF4_SetPower and ELPF4_Stop.

Appendix: Numerical Filter Design for the ELPF4

This appendix details the design procedures implemented in the design wizard and spreadsheets. The required values can be calculated easily (but roughly). For a first order approximation, the design equations are simplified to the following equations:

Equation 10

$$G = -\frac{C_1}{C_2}$$

Equation 11

$$\omega_n \omega_0 = f_{clk} \frac{\sqrt{C_2 C_3}}{\sqrt{C_A C_B}}$$

Equation 12

$$d = \frac{C_4}{\sqrt{C_A C_B}} \sqrt{\frac{C_3}{C_2}}$$

The elliptical zero for each section is calculated from the input data and the other already calculated values:

Equation 13

$$C_{PP} = \text{int}\left(C_1 C_2 C_A \left(\left(\frac{f_{sample}}{2\pi f_{zero}} + 0.25\right)^2 + 0.5\right)\right)$$

In Equation 13, CPP, C1, C2, and CA are values of capacitors.

These equations clearly show the first order interaction of the ratios of the capacitors.

The clock frequency, fclk, is calculated by rearranging Equation 11:

Equation 14

$$f_{clk} = \omega_n \omega_0 \frac{\sqrt{C_A C_B}}{\sqrt{C_2 C_3}}$$

Four-Pole Design Procedure

The design procedure is provided for both biquad parts of 4-pole filter with the same algorithm. The algorithm for one of them is described in this section.

The objective of the design procedure is to yield the highest possible fclk, for the best fidelity and least aliasing.

1. Set CA and CB to 32.
2. Set C2 to the minimum integer value, 1.
3. Set the initial value of C4 to the maximum value, 31.
4. Find the smallest value for C3 such that $C_3 \geq d^2 C_2 \frac{\sqrt{C_A C_B}}{C_4}$.
5. Revise C4 from $C_4 = d\sqrt{C_A C_B}\sqrt{C_2 / C_3}$ and round to the nearest integer.

6. Using the required gain, calculate a value for C1 from Gain: Equation 5.
7. Calculate C_{pp} from Equation 13.
8. Calculate f_{clk} from Equation 14.
9. Set analog column clocks to four times fclk. Pick an analog column clock resource, f_{sysclk}, from the selections available in PSoC Designer. See the Sample Clock section in this datasheet, for additional details on clock selection.
10. Calculate the divider by rounding to the nearest integer as follows: $$n = int\left(\frac{f_{sysclk}}{4f_{clk}} + 0.5\right)$$
11. Divide the selected system clock resource by 4n, to get the actual clock frequency.
12. Calculate the damping ratio, d, and the natural frequency, ω0, from Equation 7 and Equation 6. Compare the results to the desired values. This provides an adequate starting point for the filter design. d and ω0 will both be in error by -2 to -10% from the required values.
13. Adjust the values of C2, C3 and C4 to achieve the required values of d and ω0 using Equation 6 and Equation 7. In general, start by reducing C4 to meet the d requirement.
14. Recalculate system clock frequency to meet ω0 requirements with updated capacitor values.

The previous procedure yields approximate values for the capacitors in the ELPF4. More exact calculations, with close adherence to design equations 4, 5 and 6, are executed in the development spreadsheet ELPF4 Design. This spreadsheet is available in the PSoC Designer documentation directory. ELPF4 design is implemented in a filter wizard accessible by right-clicking on the user module in the Device Editor. After filter optimization, the capacitor values and clock frequency divisor coefficients can be entered in the PSoC Designer User Module Parameter window.

Four-Pole Example

For this design example, f-3dB is set to 1.0 kHz. For the Butterworth filter, the normalized corner frequency, ω0, is 1.0 and the damping ratio, d, is 1.414. The filter is designed for the maximum oversample ratio.

The design procedure is followed by the numbers:

1. CA, CB = 32.
2. C2 = 1.
3. C4 = 31.
4. Calculate C3 very close to 2.
5. No change to C4.
6. C1 = 1.
7. $$f_{clk} = \omega_n \omega_0 \frac{\sqrt{C_A C_B}}{\sqrt{C_2 C_3}} = 2\pi 10^3 \cdot 1 \cdot \frac{\sqrt{32 \cdot 32}}{\sqrt{2}} = 142.17\ kHz$$
8. Analog Column Clock = 568.69 kHz.
9. n = 42 from 24V1, 24V2 clock or Timer, or Counter PSoC block.

10. fclk = 142.857 kHz.

$$d = \frac{C_4\left(\frac{C_3}{C_2}\right)^{\frac{1}{2}}}{\left(C_A C_B + \frac{C_2 C_3}{4} + \frac{C_4 C_3}{2}\right)^{\frac{1}{2}}} = \frac{31\sqrt{2}}{\left(32 \cdot 32 - \frac{2}{4} - \frac{31 \cdot 2}{2}\right)^{\frac{1}{2}}} = 1.392$$

$$\omega_n \omega_0 = \frac{f_s (C_2 C_3)^{\frac{1}{2}}}{\left(C_A C_B - \frac{C_2 C_3}{4} - \frac{C_4 C_3}{2}\right)^{\frac{1}{2}}} = \frac{142.857\,\text{kHz}\,\sqrt{2}}{\left(32 \cdot 32 - \frac{2}{4} - \frac{31 \cdot 2}{2}\right)^{\frac{1}{2}}}$$

f-3dB = 1.020 kHz and ω0 = 1.00 orf-3dB = 1.000 kHz and ω0 = 1.02

11. For many applications, 1.6% error in damping ratio and 2% error in corner frequency are adequate.

12. This performance can be improved. f-3dB can be adjusted closer to the desired value by using the 48 MHz input to a Timer or Counter User Module set to divide by 85, rather than setting the system clock dividers to divide 24 MHz by 42. This yields f-3dB equal to 1.008 kHz.

Version History

Version	Originator	Description
1.1	DHA	Added CY8C24090 base device support.
1.20	DHA	1. Added help file to wizard. 2. Updated images in this user module datasheet.
1.20.b	MYKZ	Users can now store printer settings in User Module Wizard.

Note PSoC Designer 5.1 introduces a Version History in all user module datasheets. This section documents high level descriptions of the differences between the current and previous user module versions.

Copyright © 2009-2013 Cypress Semiconductor Corporation. The information contained herein is subject to change without notice. Cypress Semiconductor Corporation assumes no responsibility for the use of any circuitry other than circuitry embodied in a Cypress product. Nor does it convey or imply any license under patent or other rights. Cypress products are not warranted nor intended to be used for medical, life support, life saving, critical control or safety applications, unless pursuant to an express written agreement with Cypress. Furthermore, Cypress does not authorize its products for use as critical components in life-support systems where a malfunction or failure may reasonably be expected to result in significant injury to the user. The inclusion of Cypress products in life-support systems application implies that the manufacturer assumes all risk of such use and in doing so indemnifies Cypress against all charges.

PSoC Designer™ and Programmable System-on-Chip™ are trademarks and PSoC® is a registered trademark of Cypress Semiconductor Corp. All other trademarks or registered trademarks referenced herein are property of the respective corporations.

Any Source Code (software and/or firmware) is owned by Cypress Semiconductor Corporation (Cypress) and is protected by and subject to worldwide patent protection (United States and foreign), United States copyright laws and international treaty provisions. Cypress hereby grants to licensee a personal, non-exclusive, non-transferable license to copy, use, modify, create derivative works of, and compile the Cypress Source Code and derivative works for the sole purpose of creating custom software and or firmware in support of licensee product to be used only in conjunction with a Cypress integrated circuit as specified in the applicable agreement. Any reproduction, modification, translation, compilation, or representation of this Source Code except as specified above is prohibited without the express written permission of Cypress.

Disclaimer: CYPRESS MAKES NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Cypress reserves the right to make changes without further notice to the materials described herein. Cypress does not assume any liability arising out of the application or use of any product or circuit described herein. Cypress does not authorize its products for use as critical components in life-support systems where a malfunction or failure may reasonably be expected to result in significant injury to the user. The inclusion of Cypress' product in a life-support systems application implies that the manufacturer assumes all risk of such use and in doing so indemnifies Cypress against all charges.

Use may be limited by and subject to the applicable Cypress software license agreement.