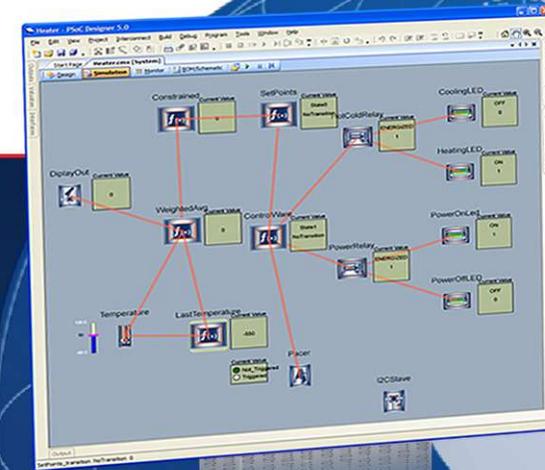


バンドパスフィルタの設計

bpf

*PSoC Experiment
Lab*

Experiment Course Material V2.20
July 2nd, 2019
bpf.pptx (56Slides)



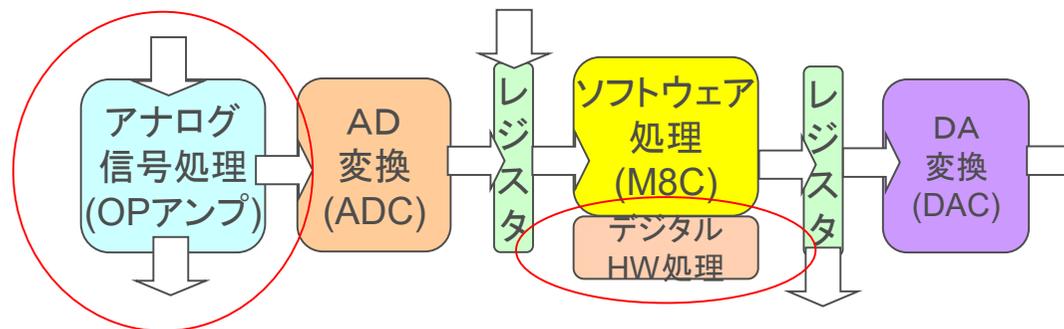
Renji Mikami
Renji_Mikami(at_mark)nifty.com



信号処理によって応用が格段に広がってきます

ラボ bpf

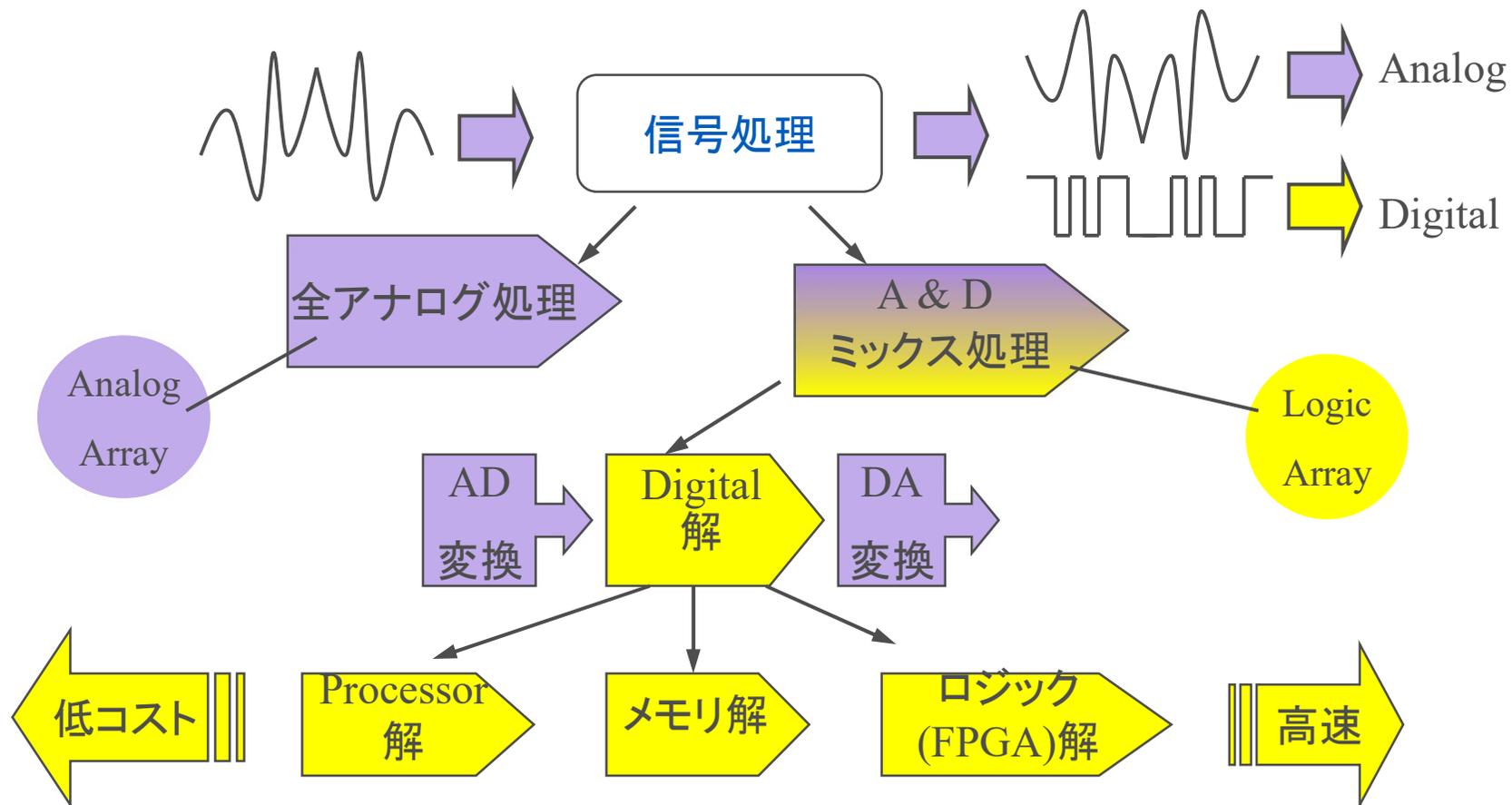
バンドパスフィルタの設計





信号処理や制御の多様なアプローチ

アナログ, デジタル・エンジン, プロセッサの3方式と直列, 並列処理, 直並列ミックス, 多様なシステム構築解がある



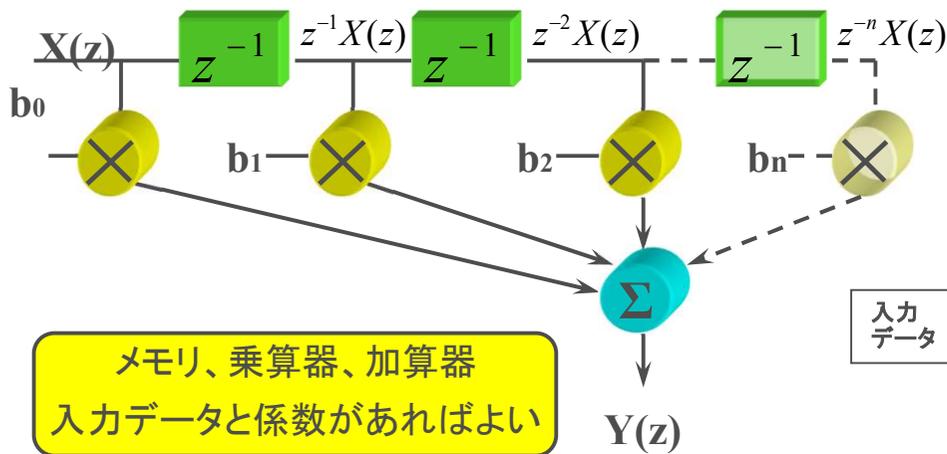
★ 信号処理のデジタル実装解

スペック

Z変換 FIR Filter一般式の例

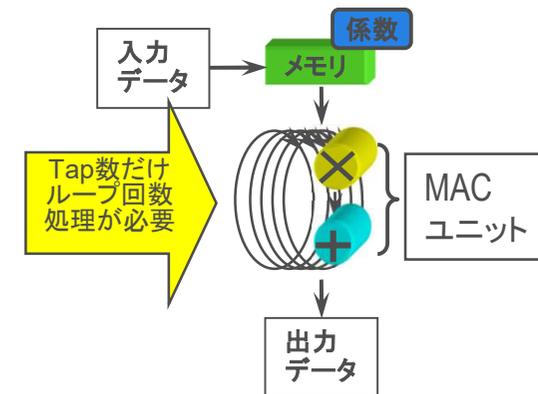
$$Y(z) = b_0 X(z) + b_1 z^{-1} X(z) + b_2 z^{-2} X(z) \dots$$

$$= (b_0 + b_1 z^{-1} + b_2 z^{-2} \dots) X(z)$$

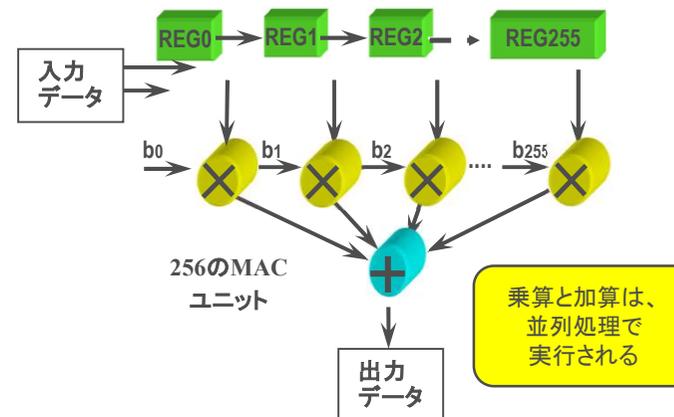


Processor 解

DS Processorの処理



ハードロジックによる並列処理





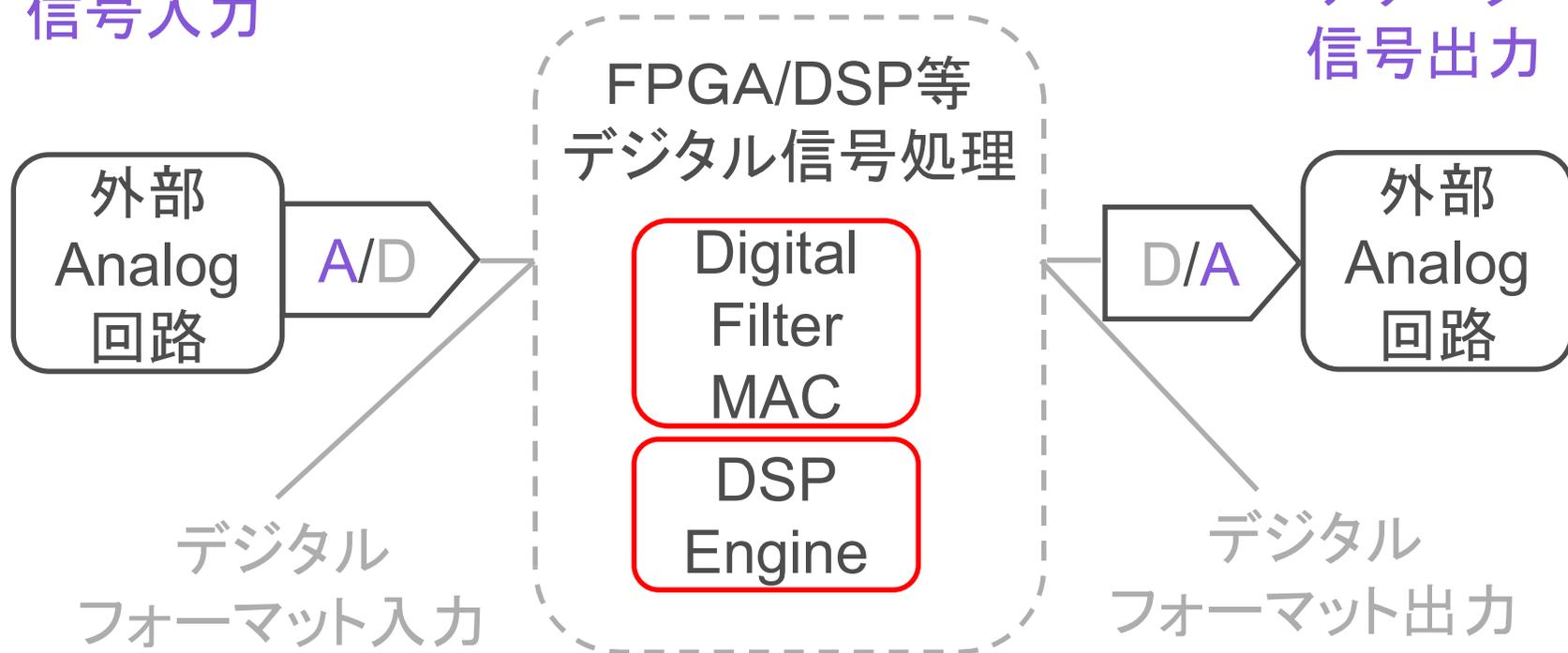
デジタル信号処理の考え方

現代のデジタル・コンバージェンスでは信号処理をA/D変換後にすべてデジタルで行う方向に向かっている.この方法は元来膨大なハードウェアのバックグラウンドと高精度のD/AやA/Dが前提となるがLSIの高集積化によって実用領域になってきたという経緯がある

アナログ
信号入力

信号処理をデジタル
化して行う

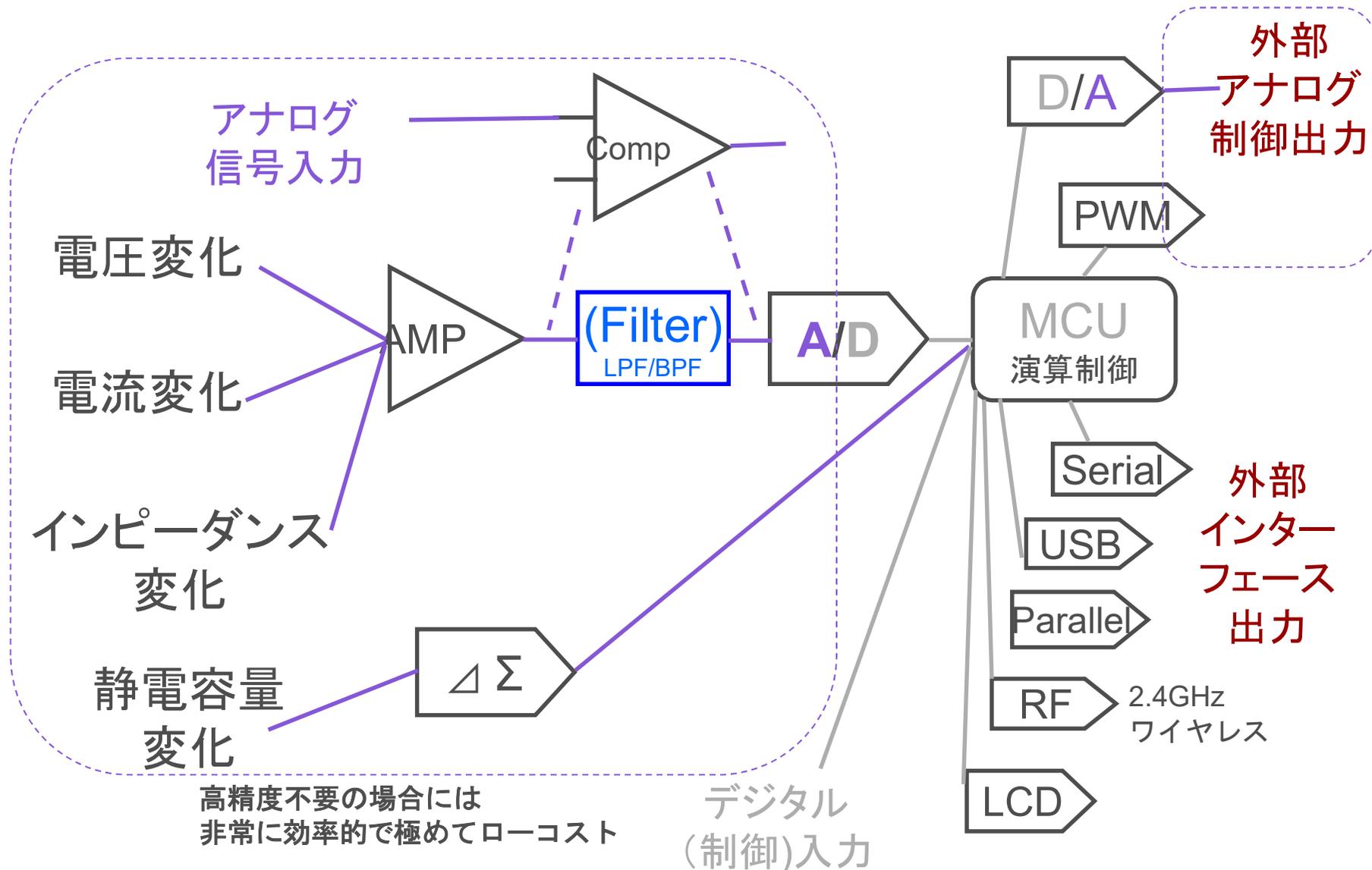
アナログ
信号出力



入口と出口はアナログが存在していることに注意



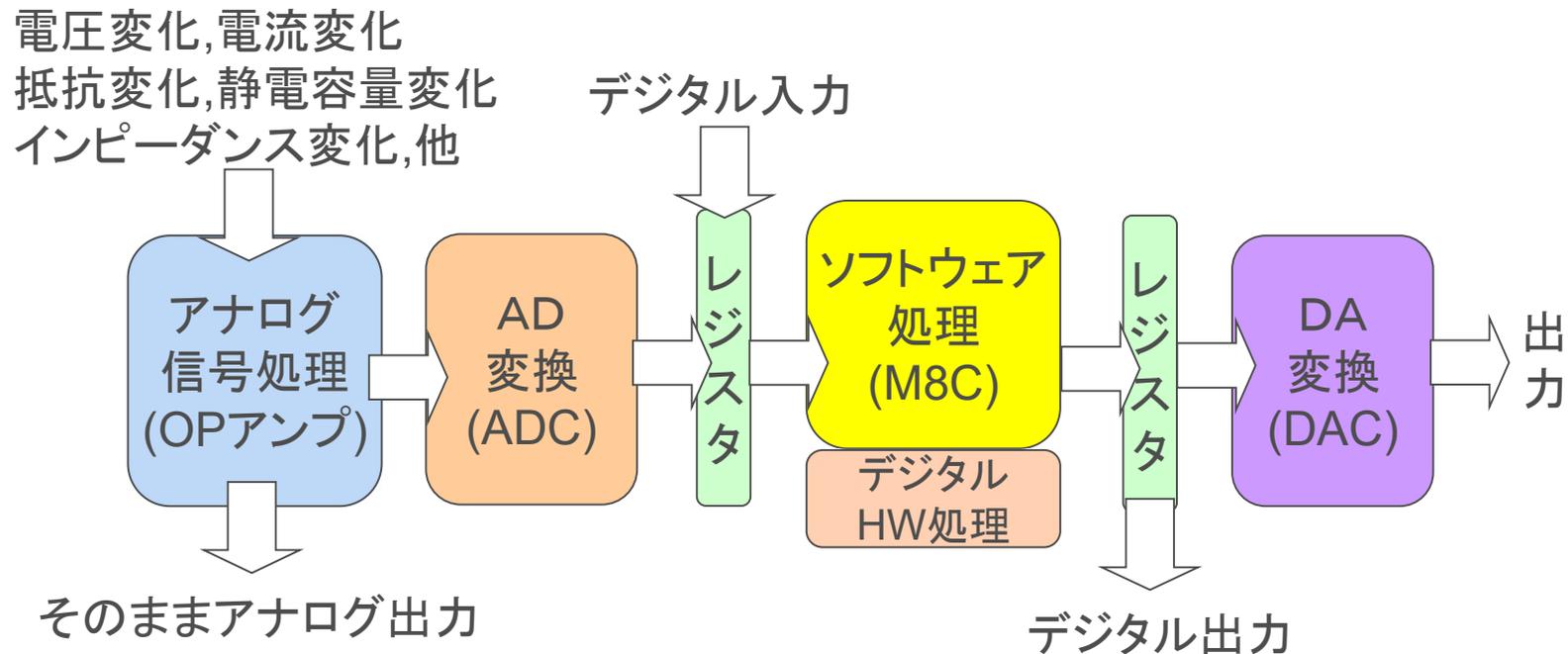
PSoCでは信号処理を前工程(アナログ処理)に移動





PSoCによる処理工程

- 外部現象変化をセンサーが電気信号に変換
- センサーの出力はアナログ信号
- 信号処理(デジタルまたはアナログ)
- AD変換してMPUで処理(レジスタがI/F)
- DA変換して外部現象を発生(レジスタがI/F)





主要な各回路ブロック-ユーザーモジュール

デジタル・アナログ入出力

GPIO (General Purpose I/O)にてデジアナ双方のI/O可能

アナログ微小信号の増幅 : PGAとINSAMP

直流増幅(演算), 交流増幅 > PGA (単電源非反転オペアンプ)

高精度な計装アンプ > INSAMP (複数PGAトポロジー)

フィルタ回路 : LPFとBPF

SC(スイッチト・キャパシタ)ブロックで実装

AD変換 : ADCINC , DA変換 : DACn

スケラブルなADコンバータ > ADCINC, DAC

ソフトウェアによる処理 : M8C

外部制御 : アナログ, デジタル

PWMn, PDM(PRS-Pseudo Random Sequence応用)

通信 : シリアル, パラレル, USB2.0, 赤外線, 2.4G無線

割り込みコントローラ

アナログ
信号処理
(OPアンプ)

AD
変換
(ADC)

ソフトウェア
処理
(M8C)

DA
変換
(DAC)

デジタル
HW処理



信号波形からスペクトル解析へ

アナログ的に変化する信号は、オシロスコープなどではレベル(Y軸)と時間(X軸)で信号波形として表示されます。

このとき信号が例えばサイン波のように繰り返して現れる(周期関数として表現できる)場合には、この繰り返し時間に帯域窓を設定して、フーリエ変換すると特定周波数のレベルを得ることができます。

しかしフーリエ変換では、窓幅(窓時間)が一定のため時間軸方向の情報は失われてしまいます。

これには窓幅を次々に変えて時間軸方向のデータを連続的に取得して計算すればよいわけです。しかし実際には適正サンプル数とレベルを得るためにはこの窓幅を低い周波数では広く、高い周波数では狭くしていく必要があります。

このような変換を行うにはウェーブレット変換を使用します。これが今日のデジタル信号処理や通信の基礎になっています。非常に面白いですから興味のある方はこれをキーワードに調べてみましょう。

PSoCでもFFTやウェーブレット変換でデジタル信号処理の実験ができます。スペクトル解析のための基礎となります。

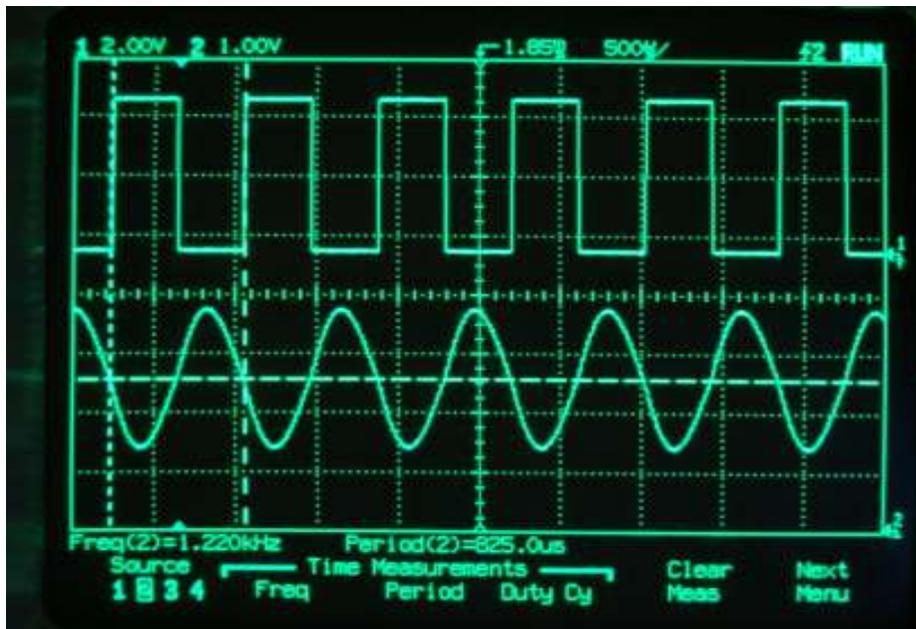


方形波をBPFに通す

方形波はその基本周波数の10倍程度の高調波成分を含んでいます。これは奇数次の周波数のサイン波で合成できます(逆FFT)。

よって方形波をBand Path Filterに通すとサイン波が得られます。

FFTを数式レベルで理解していても実際に基底周波数のサイン波がBPFによって現れることを実験で確かめることができます。PSoCには Switched Capacitor BlockでLPF, BPFユーザーモジュールを構成します。特性の設定もパラメタライズされています



PWMで作った方形波

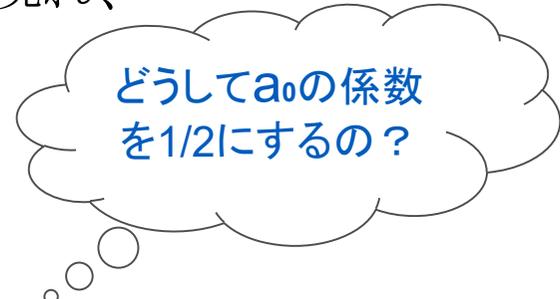
BPFフィルタリングして
現れた基底周波数の
サイン波

フーリエ級数式

$$\begin{aligned}
 f(x) &= a_0 \cos x \\
 &+ a_1 \cos x + b_1 \sin x \\
 &+ a_2 \cos 2x + b_2 \sin 2x \\
 &+ a_3 \cos 3x + b_3 \sin 3x \dots\dots + a_n \cos nx + b_n \sin nx
 \end{aligned}$$

x は、最も低い周波数で基底(基本)波といいます。
 他の項には、すべて基底波の整数倍の波が現れ、
 これらを高調波といいます。

a_0 を $\frac{1}{2}a_0$ と置き直してまとめ(1)式を得ます。

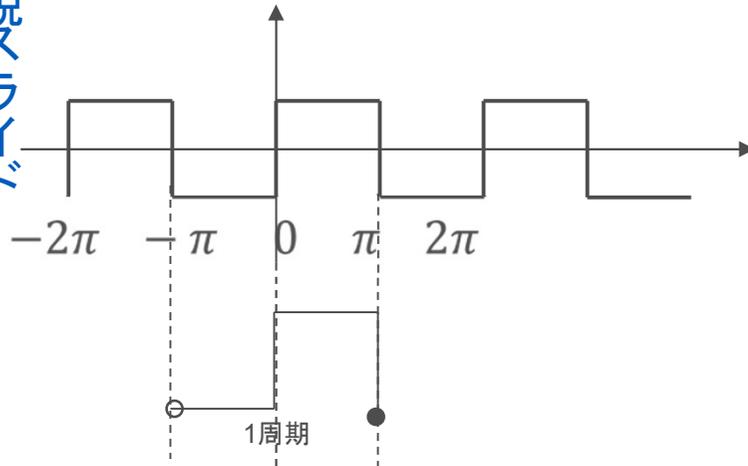


$$f(x) = \frac{1}{2}a_0 + \sum_{n=1}^{\infty} (a_n \cos nx + b_n \sin nx) \dots\dots (1)$$

$\frac{1}{2}a_0$ は、定数項(直流成分)、 $\frac{1}{2}$ はノルムが倍のため、係数合わせ
 $f(x)$ は、級数のため非連続値です。

矩形波をオシロ/FFTアナライザで観測する

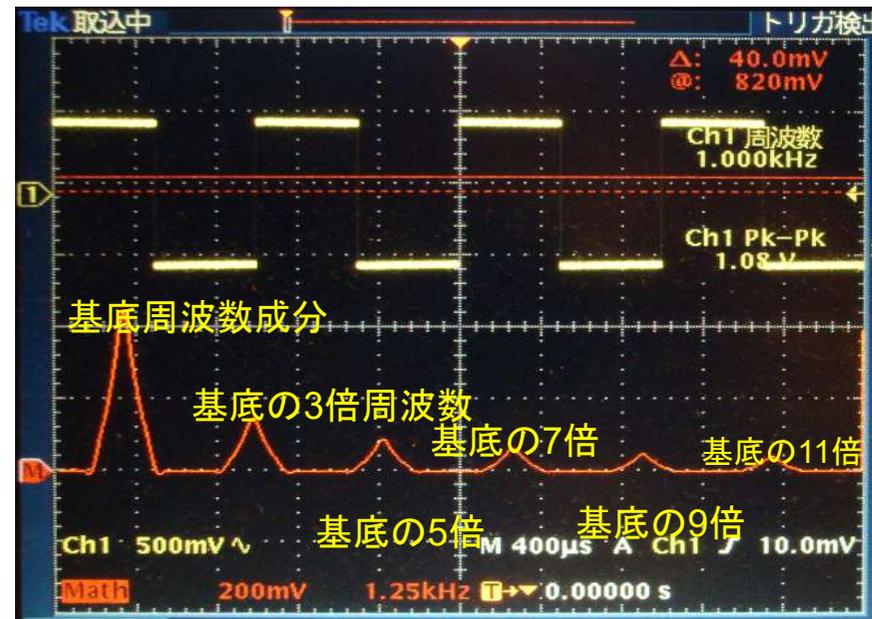
解説
スライド



$$f(x) = \sin x + \frac{1}{3} \sin 3x + \frac{1}{5} \sin 5x + \frac{1}{7} \sin 7x \dots$$

フーリエ級数展開の詳細などは、
3年ComputerScience Aでやります。

上は、矩形波の時間軸波形(横軸は時間)
下は、スペクトル(振幅はRMS)(横軸は周波数)



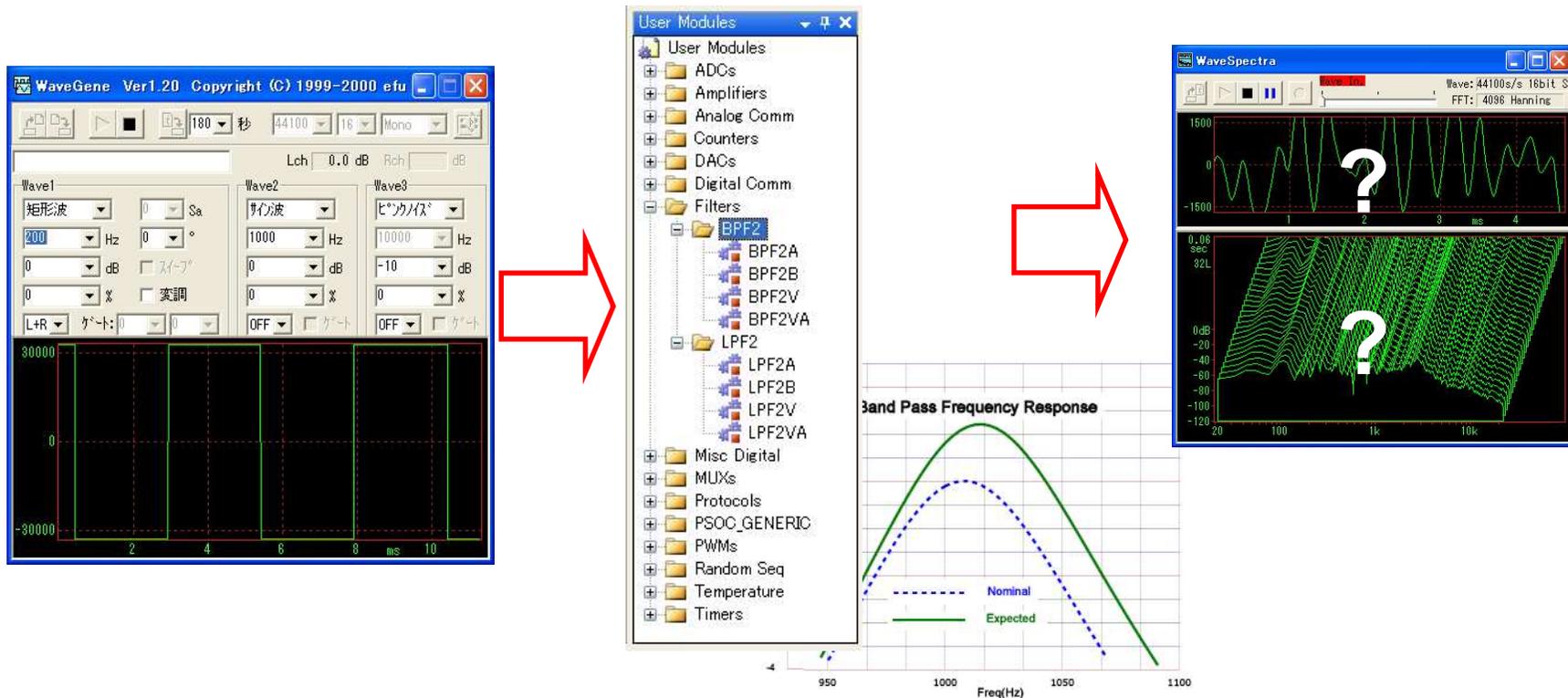
基底波を1KHzとすると
1KHz, 3KHz, 5KHz, 7KHz

なぜcosの項が現れないの
でしょうか？



方形波から正弦波をとりだす

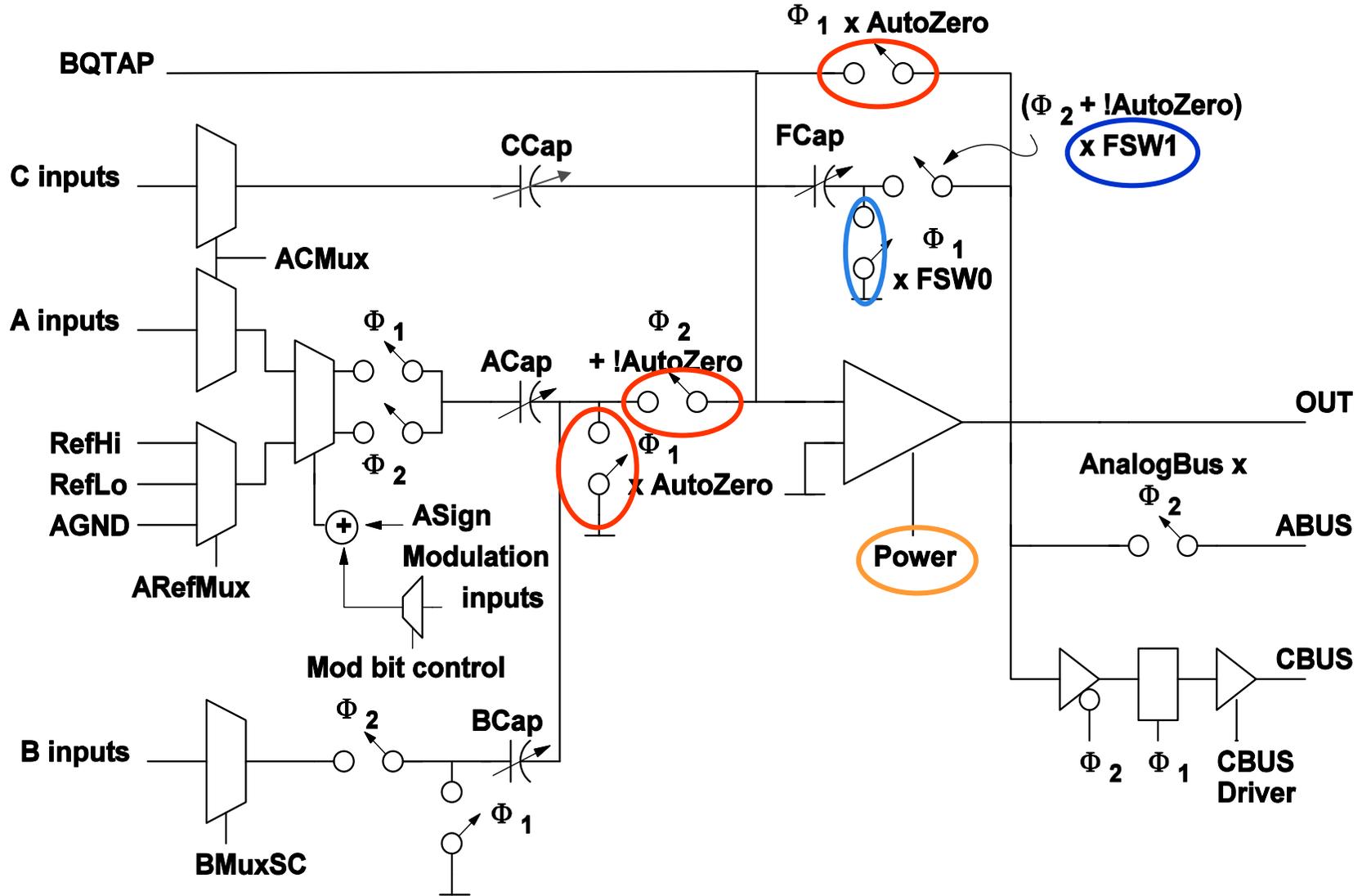
方形波信号をPSoCのバンドパス・フィルターにかける
特定スペクトラムの周波数だけをパスさせる
音を聴いてみる。口笛の音は正弦波に近い





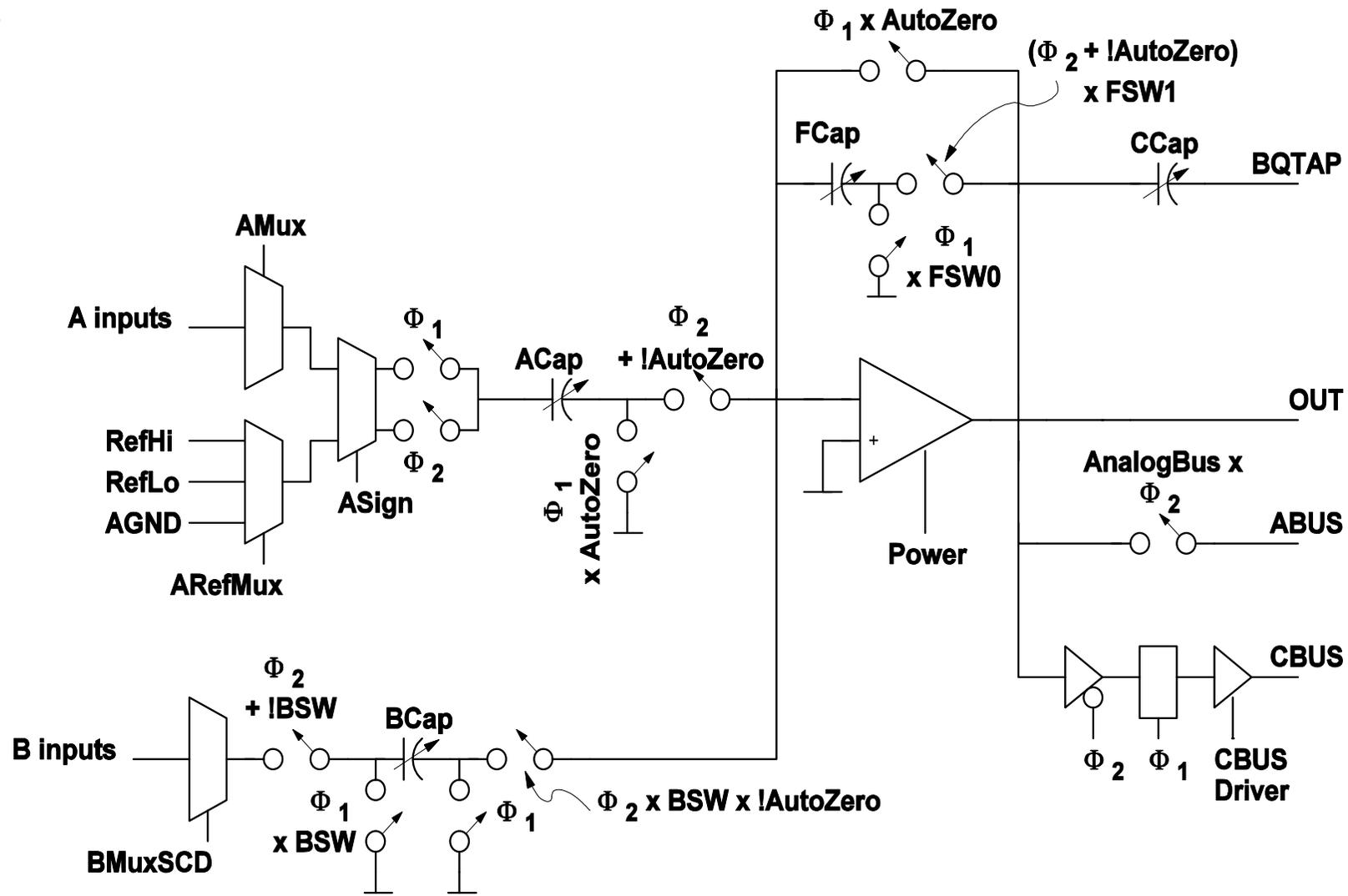
解説スライド

PSoC スイッチトキャパシタ **Type C** Block





Type D Switched Capacitor Blocks



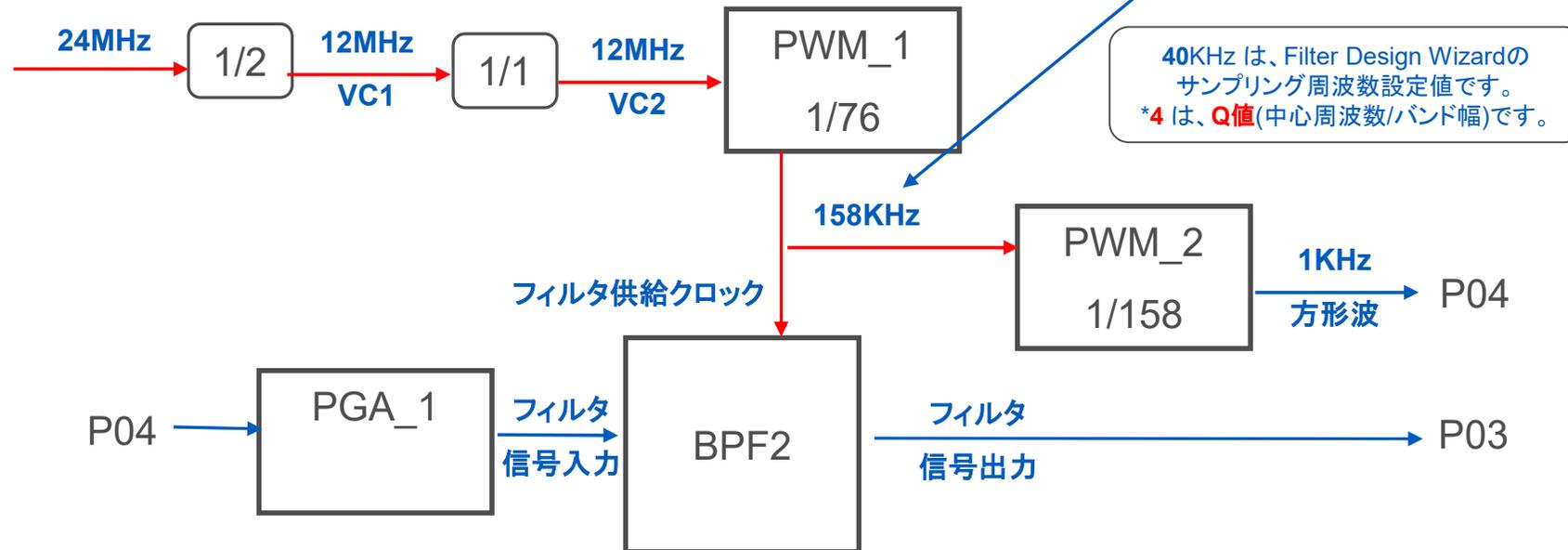


全体の構成

解説スライド

使用するユーザーモジュール
PWM8 x 2, PGA, BPF2

BPFモジュールに与えるクロック周波数は、
計算上 $40\text{KHz} \times 4 = 160\text{KHz}$ ですが、
近似で 158KHz を与えています。



40KHz は、Filter Design Wizardの
サンプリング周波数設定値です。
*4 は、Q値(中心周波数/バンド幅)です。

フィルタウィザードを使用してSC(Switched Capacitor)の設定値を計算してBPFに設定します.続いてPWMを使用してSCフィルタ部に与えるクロックを生成します.検証用にPWM_2から1KHzの方形波を生成してPGAでゲインを調整してBPFに入力してサイン波が出るかチェックします.そのあとPWM_1の出力クロックを変更してみます.



フィルタ周波数1KHz,バンド幅250Hz, $F_{\text{Sample}}=40\text{KHz}$ で設計します

解説
スライド

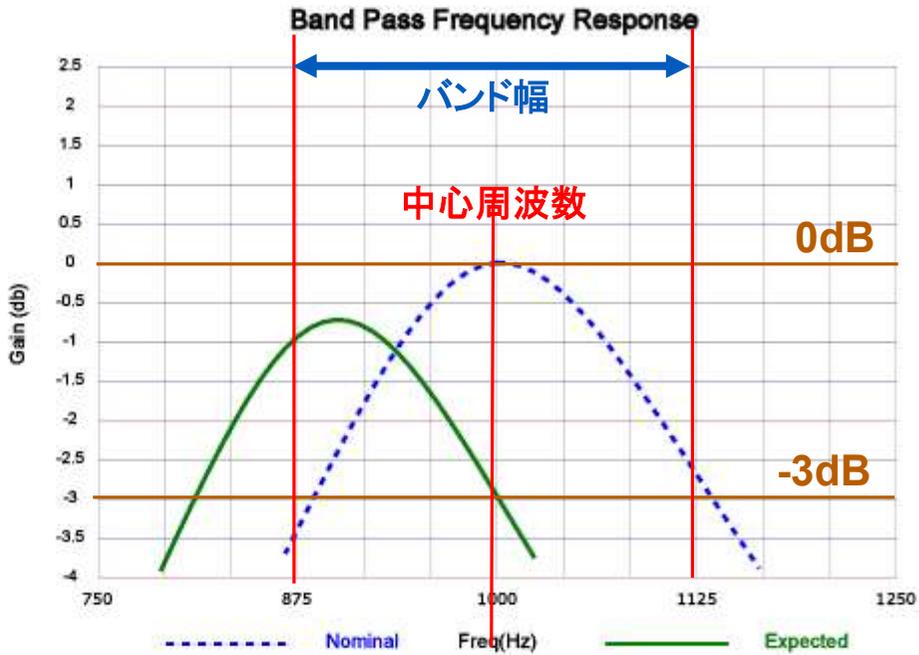
Filter Parameters

F_{Center} (Hz): 1000 Gain (dB): 0.00
 F_{Sample} (Hz): 40000 (6000 to 80000) C2: 10 (1 to 16)
 Bandwidth (Hz): 250 (40 to 2000)

バンド幅とは中心周波数の振幅から-3dBレベル低下した周波数の幅のことです。

Calculated Values

Q: 4
 Calculated Q: 3.93
 Scaled f_0 : 1002
 Gain (V/V): 1
 Calc Gain (C1/C2): 0.889
 C1: 1
 C3: 2
 C4: 18
 CA: 32
 CB: 32
 Divide by n: 150
 Sample Clock (Hz): 40000
 Oversample Ratio: 40



グラフ内の
青の点線は
計算値です。
緑の実線は
予測される
性能です。
ここでは
計算値を
使用します

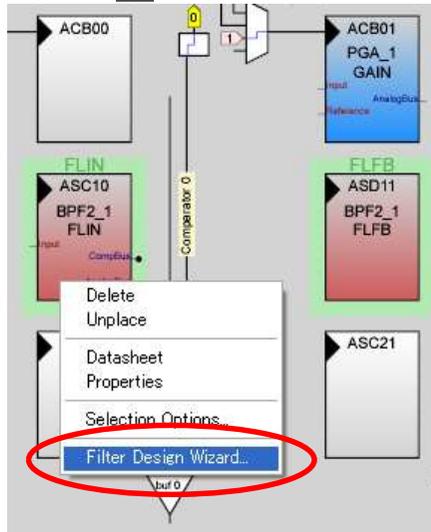
ここで $Q=1000\text{Hz}/250\text{Hz}=4$ となります。BPFモジュールに与えるクロック周波数は、 $40\text{KHz} \times 4=160\text{KHz}$ となります。このクロックはPWMで作ります。



フィルター設計ウィザード(旧版ソフトの場合)

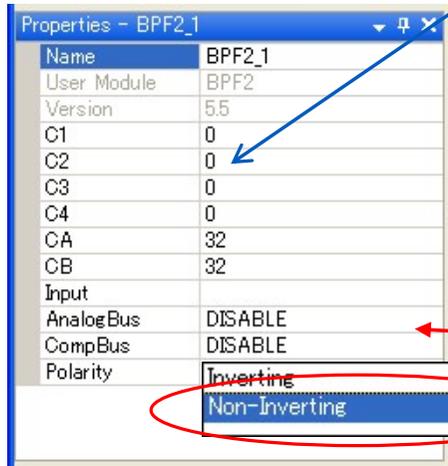
解説スライド

BPF2_1を右クリックしてFilter Design Wizard を起動

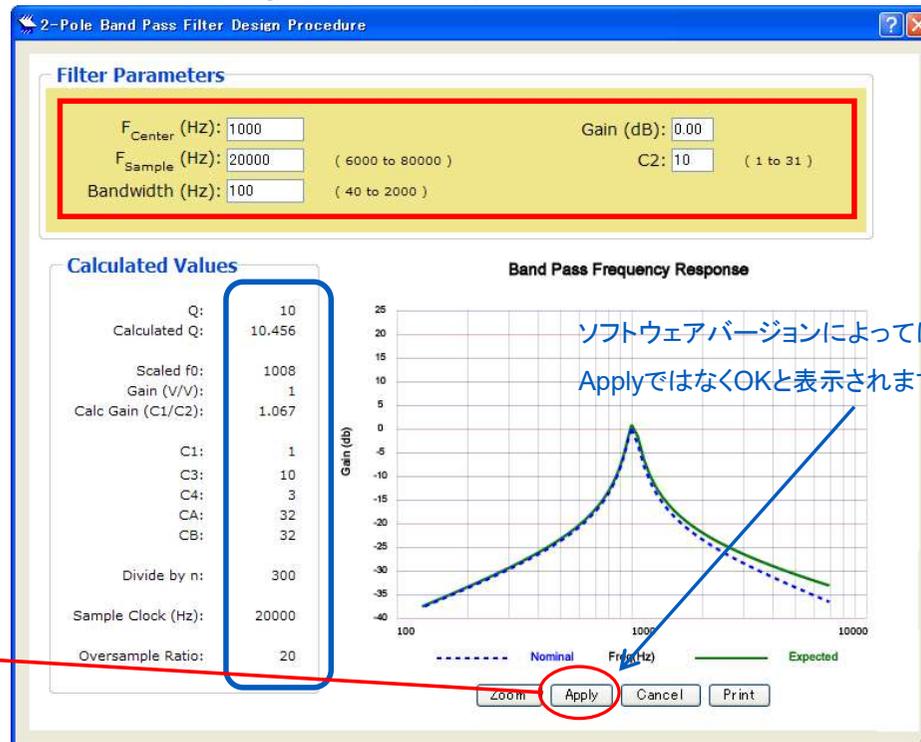


Wizard が表示されない場合は、C2の値に

10を入れてみてください



Wizard Windowの**パラメータボックス**に設計値を入力すると設定値が自動計算されます。**OK**をクリックするとプロパティウインドウに値が設定されます。複雑なSCの値計算が自動的に行われます。Zoomをクリックするとグラフが拡大されます。



ソフトウェアバージョンによってはApplyではなくOKと表示されます

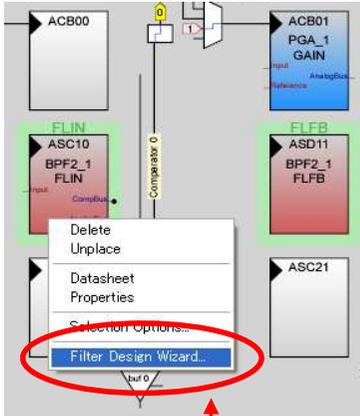
Polarity はNon-Invertingに設定



解説スライド

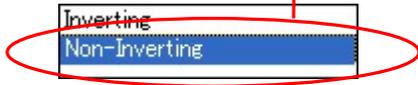
フィルター設計ウィザード(新版ソフトの場合)

BPF2_1を右クリックしてFilter Design Wizard を起動



Polarity は

Non-Invertingに設定



BPF2_1を右クリックして、Filter Design Wizard を選択してください。

Wizard Windowのパラメータボックスに設計値を入力すると設定値が自動計算されます。Apply(OK)をクリックするとプロパティーウインドウに値が設定されます。複雑なSCの値計算が自動的に行われます。新版ソフトではzoom 機能は削除されています。

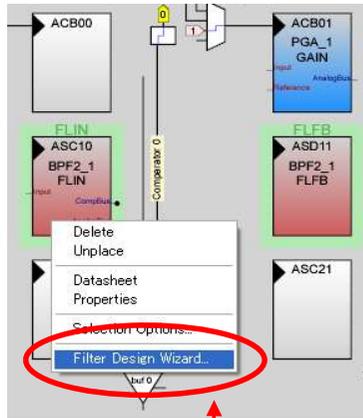
ソフトウェアバージョンによってはApplyではなくOKと表示されます



解説スライド

フィルター設計ウィザード(新版ソフトの場合)

BPF2_1を右クリックしてFilter Design Wizard を起動



Polarity は

Non-Invertingに設定



Apply (OK) をクリックすると設定終了でウィンドウが閉じますから、再度設定を見るときは、BPF2_1を右クリックして、Filter Design Wizard を選択してください。

Wizard Windowの**パラメータボックス**に設計値を入力すると**設定値が自動計算**されます。**Apply(OK)**をクリックするとプロパティウィンドウに値が設定されます。複雑なSCの値計算が自動的に行われます。新版ではzoom 機能は削除されています。

2-Pole Band Pass Filter Design Procedure

Filter Parameters

F_{Center} (Hz): 1000 Gain (dB): 0

F_{Sample} (Hz): 8000 (6000 to 80000) C2: 10 (1 to 31)

Bandwidth (Hz): 250 (40 to 2000)

Calculated Values

Q: 4.00

Calculated Q: 4.22

Scaled f₀: 1054.78

Gain (VM): 1.000

Calc Gain (C1/C2): 1

C1: 4

C3: 31

C4: 4

CA: 32

CB: 32

Divided by n: 750

Sample Clock (Hz): 8000

Oversample Ratio: 8

Band Pass Frequency Response

Gain (dB)

100 1,000 10,000

Freq (Hz)

Legend: Nominal (blue), Expected (orange)

Scale: Linear Scale Log Scale

Buttons: Print, Apply, Cancel

ソフトウェアバージョンによってはApplyではなくOKと表示されます



設計値を変更してみる(旧版ソフトのGUI)

F_{Sample}, C2値を変更するとフィルタ特性グラフが変わります
SCの値は自動的に計算されます。

バージョンによっては、パラメータの並びが異なる場合がありますから、正しい項目を選んで入力してください。

Filter Parameters

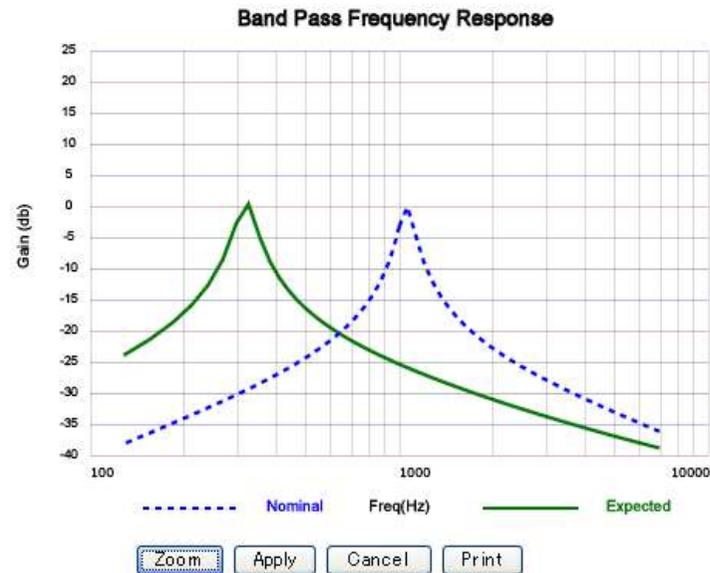
F_{Center} (Hz): 1000 Gain (dB): 0.00

F_{Sample} (Hz): **8000** (6000 to 80000) **C2: 2** (1 to 31)

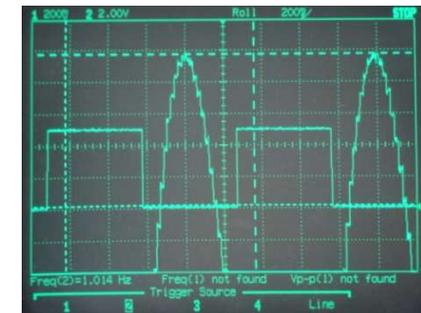
Bandwidth (Hz): 100 (40 to 2000)

Calculated Values

Q:	10
Calculated Q:	8.004
Scaled f0:	1054
Gain (V/V):	1
Calc Gain (C1/C2):	1.032
C1:	1
C3:	31
C4:	1
CA:	32
CB:	32
Divide by n:	750
Sample Clock (Hz):	8000
Oversample Ratio:	8



F_{Sample}
 サンプリング周波数を上げるほど、フィルタリング後の波形はスムーズになります。
 このオーバーサンプルレートが低いと波形には段々がつきます。

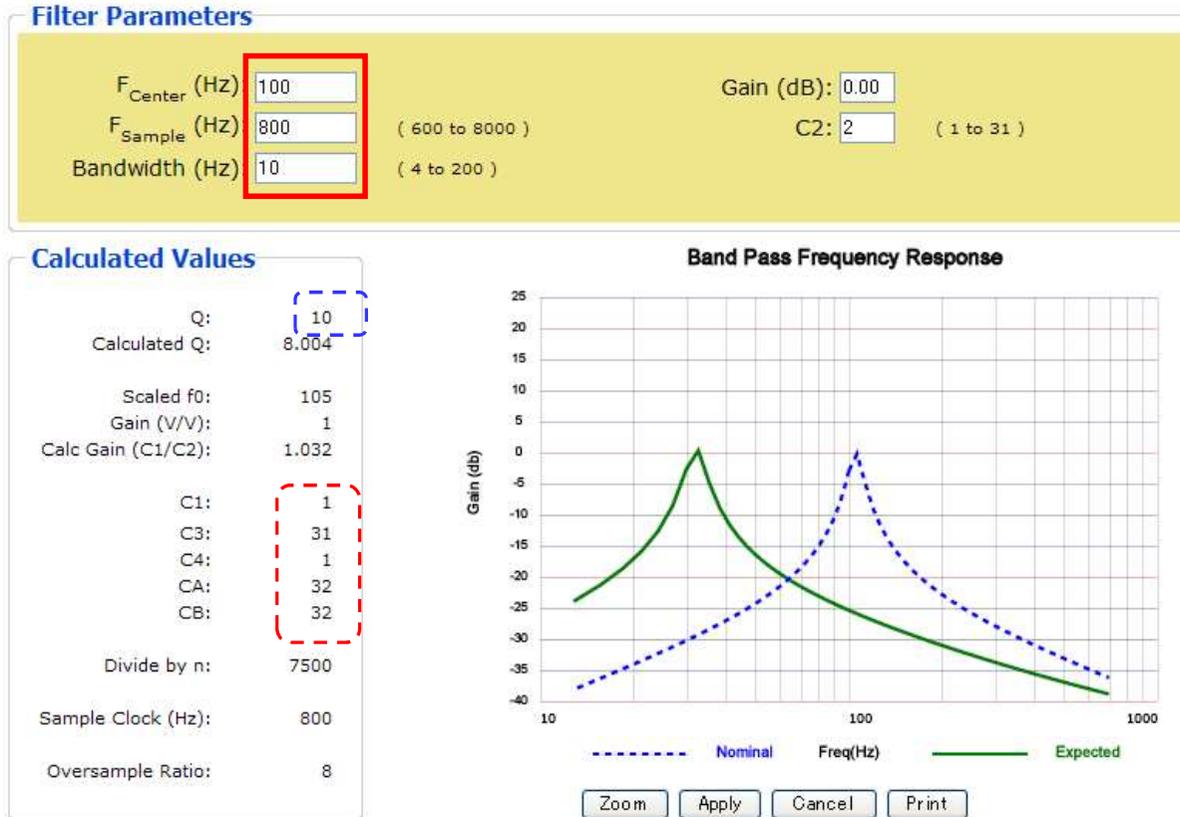




サンプリング周波数の変更の意味(旧版ソフトのGUI)

解説
スライド

F_{Center} , F_{Sample} , Bandwidth値をすべて1/10にしてみます



SCFでは
Cの定数を
そのままに
しておいて
サンプリング
周波数を変更
するだけで、
フィルタ周波数
を変更すること
ができます。
Qも変わりません!

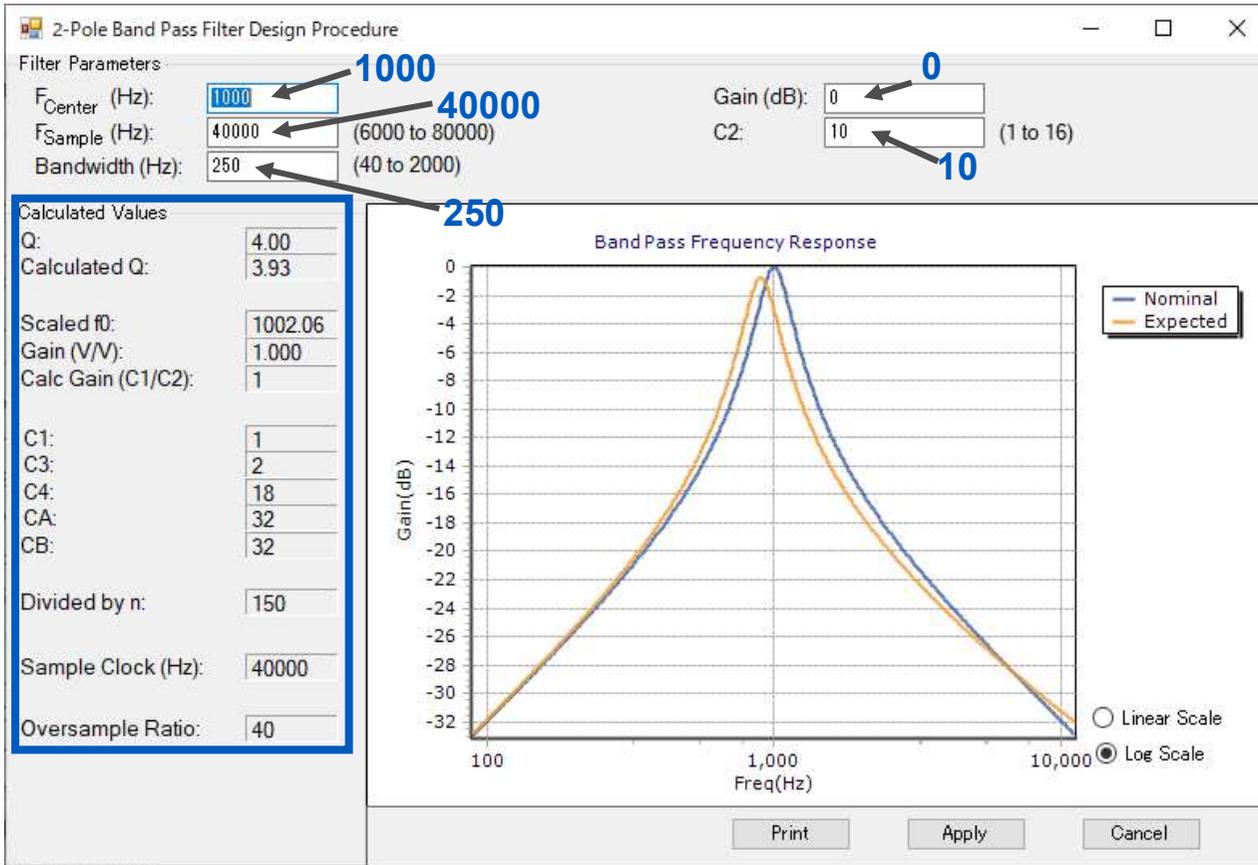
サンプルクロック,中心周波数が1/10になりましたがSCの設定値には変化はありません. このことは何を意味するのでしょうか?



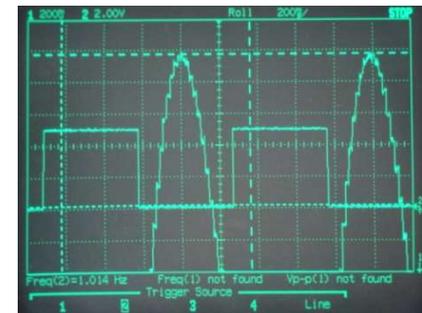
設計値を変更してみる(新版ソフトのGUI)

F_{Sample}, C₂値を変更するとフィルタ特性グラフが変わります
SCの設定値は自動的に計算されます。

バージョンによっては、パラメータの並びが異なる場合がありますから、正しい項目を選んで入力してください。



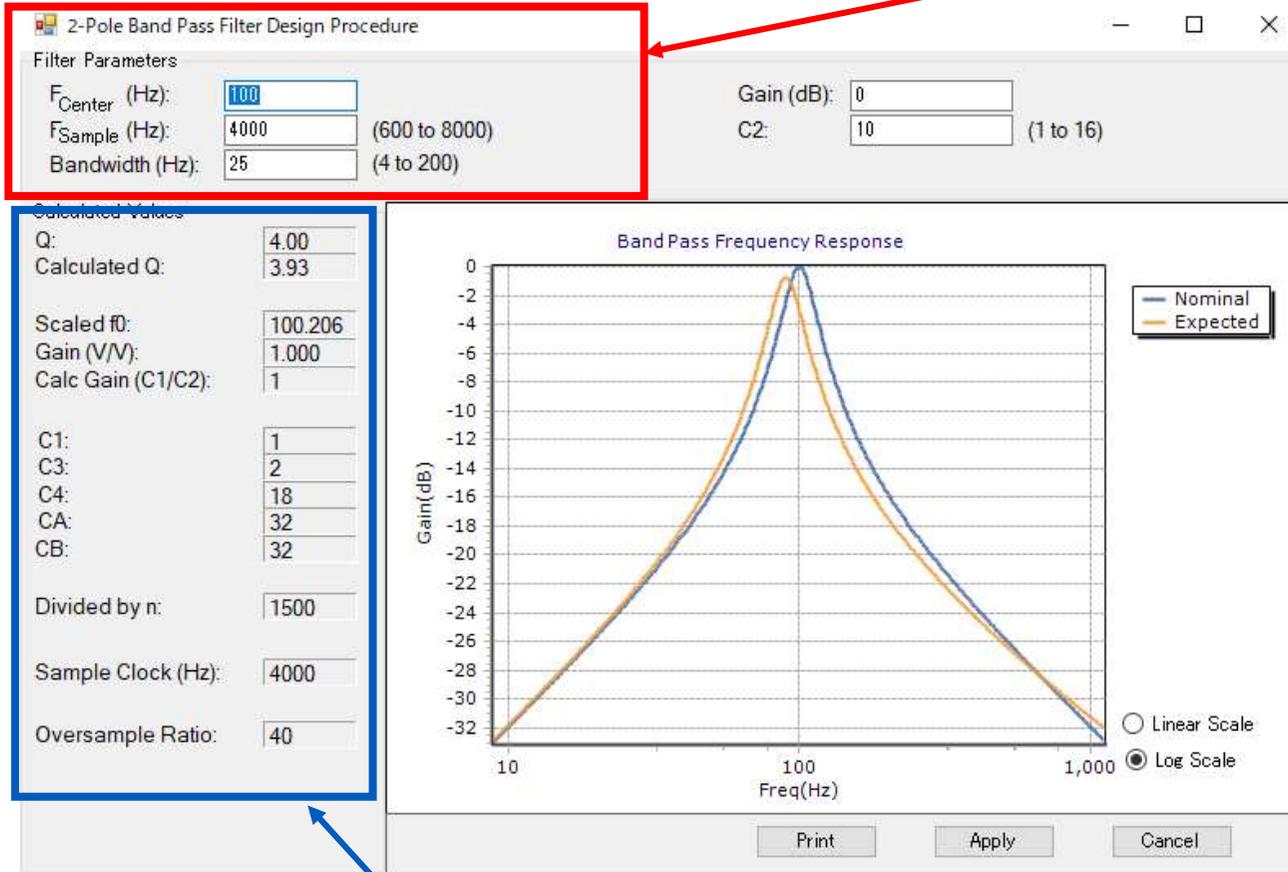
F_{Sample}
 サンプリング周波数
 を上げるほど、
 フィルタリング後の
 波形はスムーズ
 になります。
 このオーバー
 サンプルレートが
 低いと波形には
 段々がつきます。





サンプリング周波数の変更の意味(新版ソフトのGUI)

F_{Center}, F_{Sample}, Bandwidth値をすべて1/10にしてみます



SCFでは
Cの定数を
そのままに
しておいて
サンプリング
周波数を変更
するだけで、
フィルタ周波数
を変更すること
ができます。
Qも変わりません!

サンプルクロック, 中心周波数が1/10になりましたがSCの設定値には変化はありません. このことは何を意味するのでしょうか?



SC定数が変わらないということ

サンプルクロック,中心周波数が1/10になりましたがSCの設定値には変化はありません.このことは何を意味するのでしょうか?

BPFブロックに与えるクロック,つまりサンプリング周波数を変更するだけでBPFのフィルター周波数を変更できるということになります.BPFのフィルター周波数はPWMで作っていましたね.PWM周波数はプログラムで自由に変更できました.

プログラムで連続的なフィルター周波数変更が可能ということになります.ここからどんな応用が考えられるのでしょうか.



1KHz, BW250Hz, $F_{\text{Sample}}=40\text{K}$ で設計を行います(旧版ソフトのGUI)

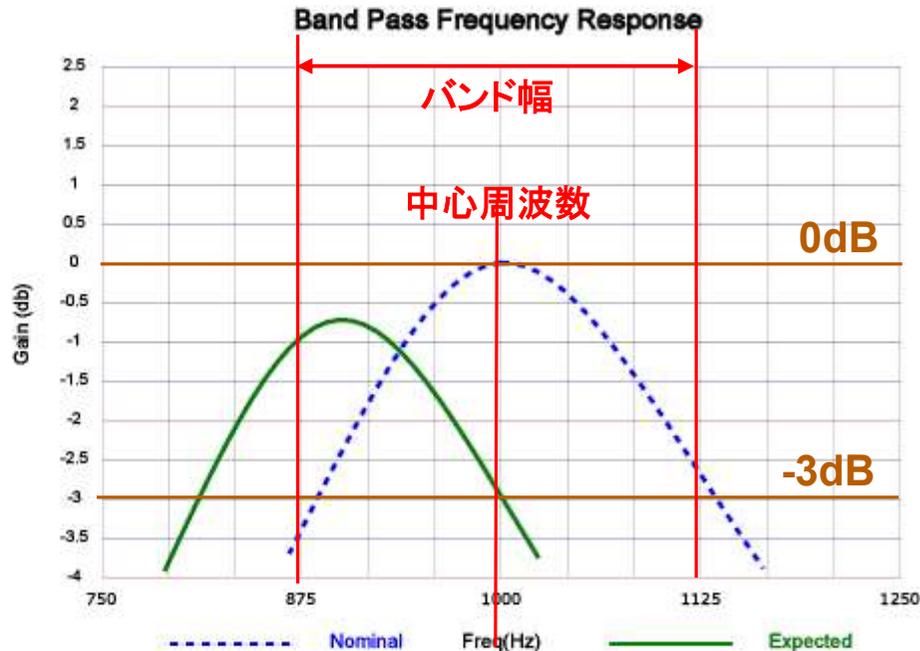
解説スライド

Filter Parameters

F_{Center} (Hz):	1000	Gain (dB):	0.00
F_{Sample} (Hz):	40000 (6000 to 80000)	C2:	10 (1 to 16)
Bandwidth (Hz):	250 (40 to 2000)		

Calculated Values

Q:	4
Calculated Q:	3.93
Scaled f_0 :	1002
Gain (V/V):	1
Calc Gain (C1/C2):	0.889
C1:	1
C3:	2
C4:	18
CA:	32
CB:	32
Divide by n:	150
Sample Clock (Hz):	40000
Oversample Ratio:	40



グラフ内の
青の点線は
計算値です。
緑の実線は
予測される
性能です。
予測性能と
実測値を
あとで比較
してみます

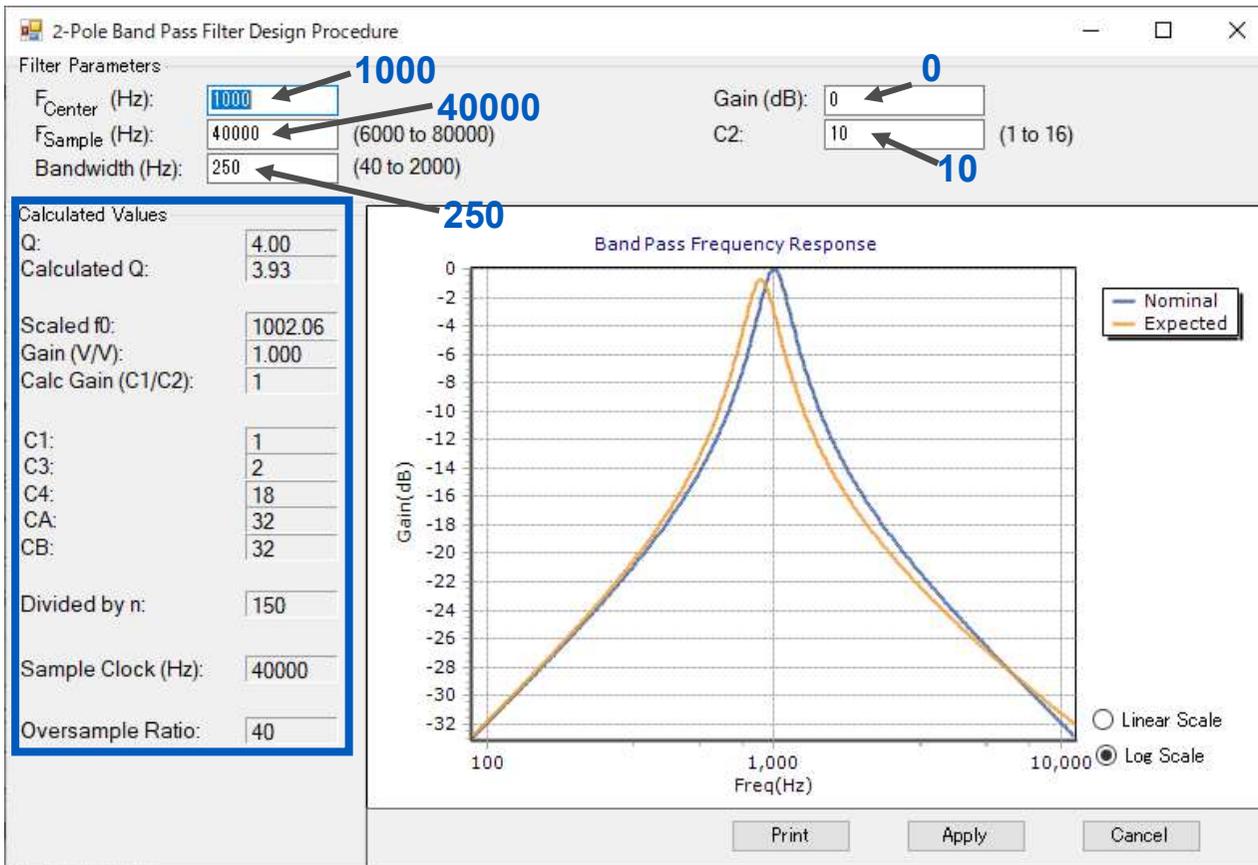
ここで $Q=1000\text{Hz}/250\text{Hz}=4$ となります。BPFモジュールに与えるクロック周波数は、 $40\text{KHz} \times 4=160\text{KHz}$ となります。このクロックはPWMで作ります。設定を確認したら **APPLY** をクリックしてPWMの設計にすすみます。



1KHz, BW250Hz, $F_{\text{Sample}}=40\text{K}$ で設計を行います(新版ソフトのGUI)

F_{Center} , F_{Sample} , Bandwidth, C2値を入力してください。
SCの設定値は自動的に計算されます。

バージョンによっては、パラメータの並びが異なる場合がありますから、正しい項目を選んで入力してください。



F_{Sample}

サンプリング周波数

を上げるほど、

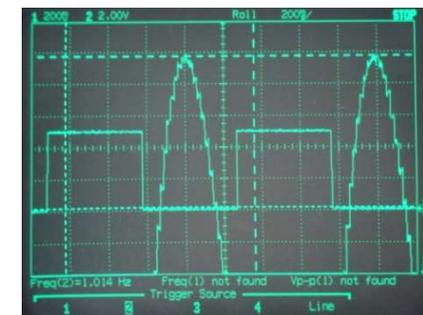
フィルタリング後の
波形はスムーズ

になります。

このオーバー

サンプルレートが

低いと波形には
段々が付きます。



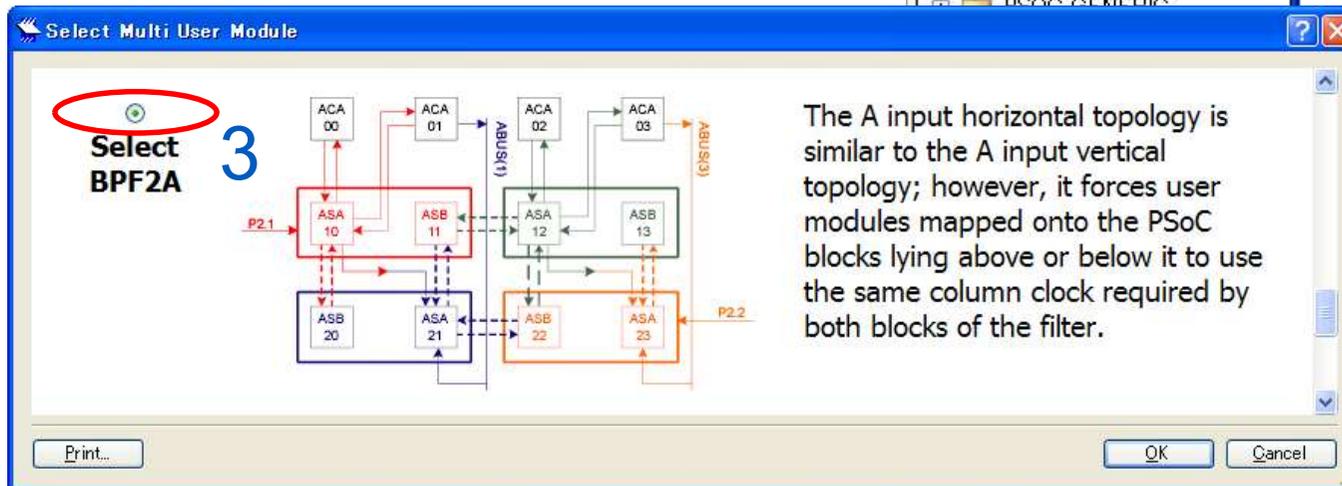
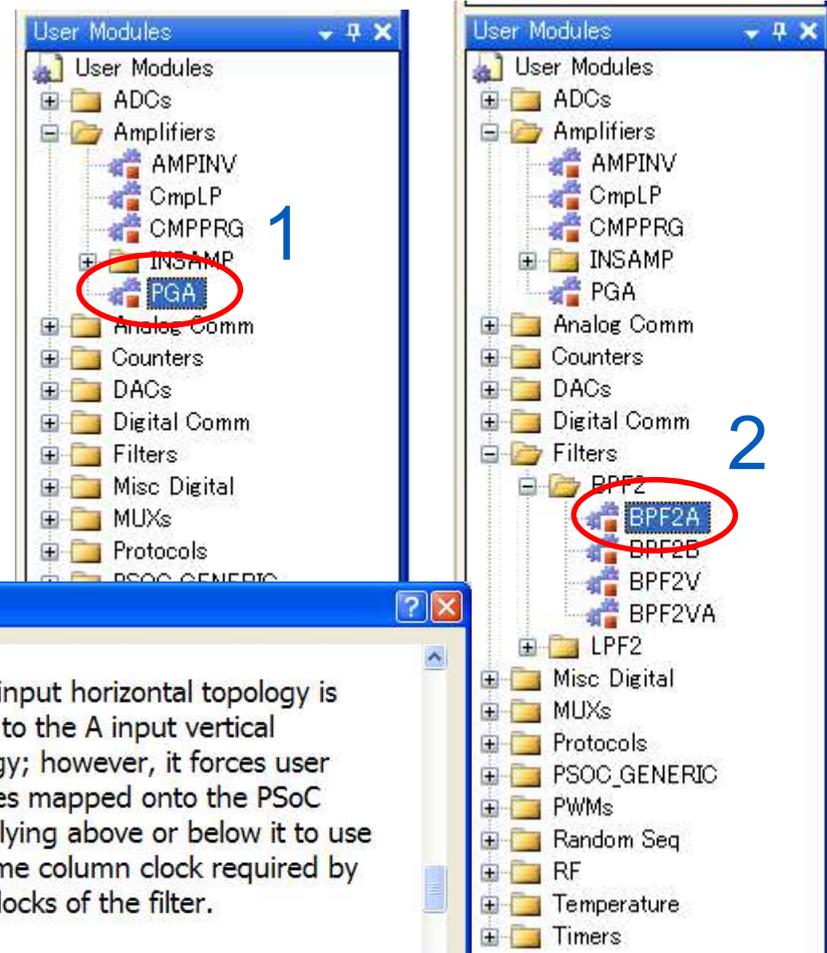
新規プロジェクトを作成

1. File > New Project
2. Chip-level Project を選択
3. Name に bpf
4. プロジェクト名を入力
5. Device 27443 選択
6. C言語を選択



ユーザーモジュール選択と配置

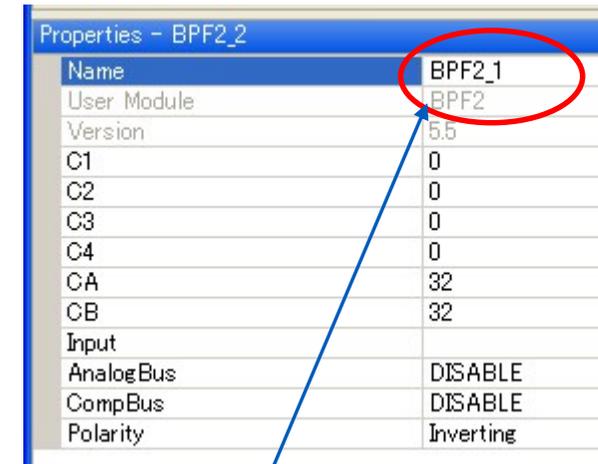
- 1.ユーザーモジュールからPGAを選択して配置
- 2.BPF2Aを選択して配置
- 3.BPF2Aの配置パターンはBPF2Aを選択



PGAとBPFの配置

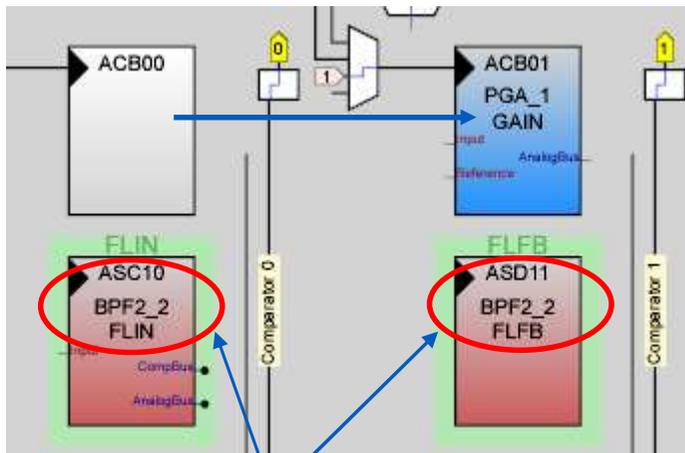
PGA 1は、自動配置された場所からひとつ右のACB01に移動してください。

BPF2は、自動配置された場所が最も下の段のブロックになった場合は以下の図の場所ASC10とASD11に配置場所を変更してください。移動しない場合は、BPFをもうひとつ配置してから最下段のBPFを削除して真ん中の段のBPF2_2を残し右クリックしてBPF2_1にRenameしてください



Properties - BPF2_2	
Name	BPF2_1
User Module	BPF2
Version	5.5
C1	0
C2	0
C3	0
C4	0
CA	32
CB	32
Input	
AnalogBus	DISABLE
CompBus	DISABLE
Polarity	Inverting

BPF2_1に
変更してください

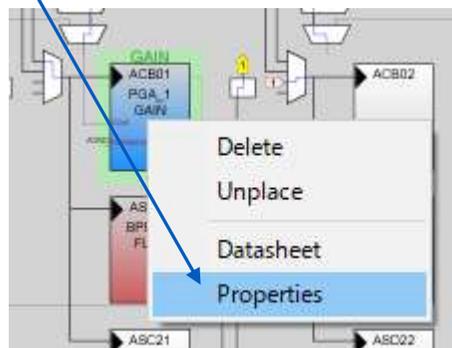


上図の表示はBPF2_2ですがBPF2_1に変更されます

左FLIN(Filter IN)はASC10, 右FLFB(Filter Feedback)はASD11ブロックを使用していますがSCブロックにはC型とD型があります。

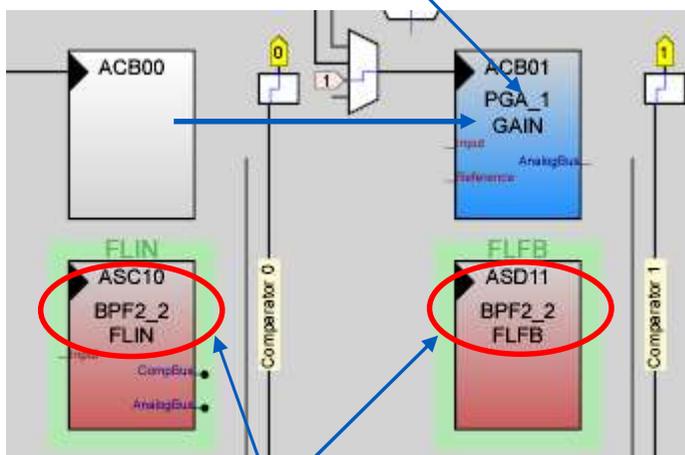
PGA_1の設定の確認

PGA_1を右クリックして、Propertiesを選んで開き、設定を確認してください。



PGA_1 の設定

Parameters - PGA_1	
Name	PGA_1
User Module	PGA
Version	3.2
Gain	0.750
Input	AnalogColumn_InputSelect_1
Reference	AGND
AnalogBus	Disable

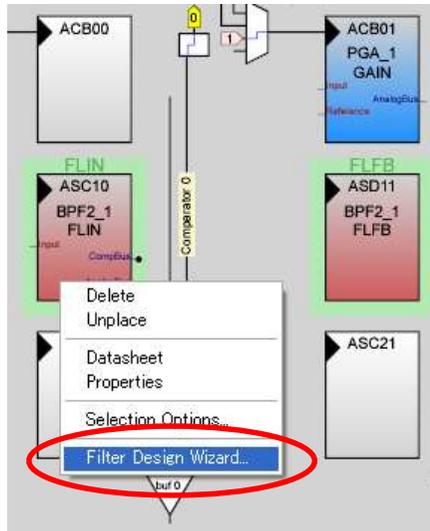


上図の表示はBPF2_2ですがBPF2_1に変更されます

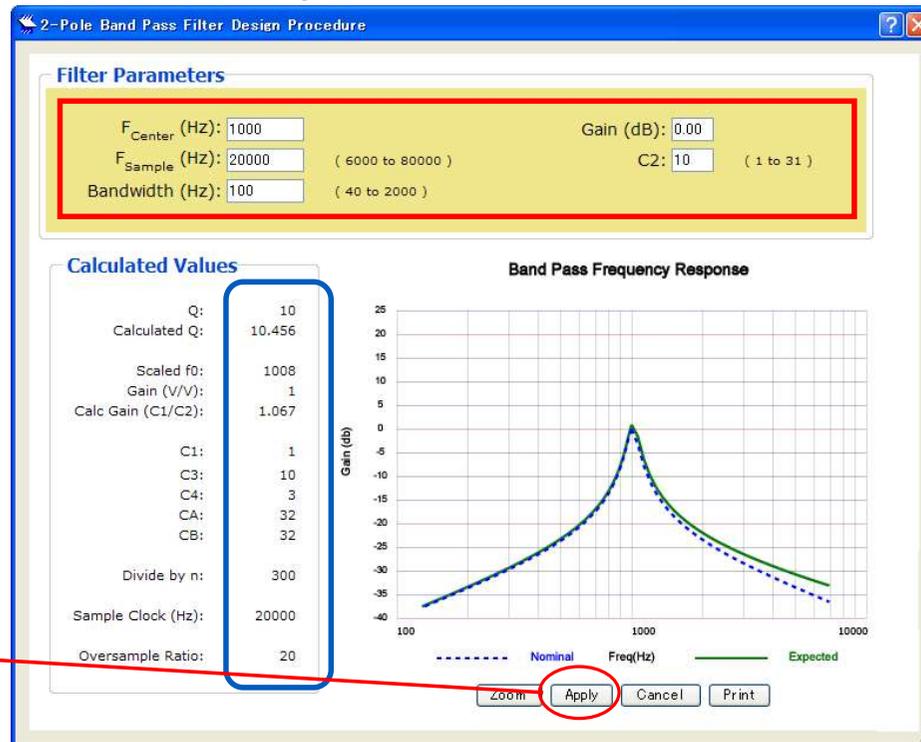
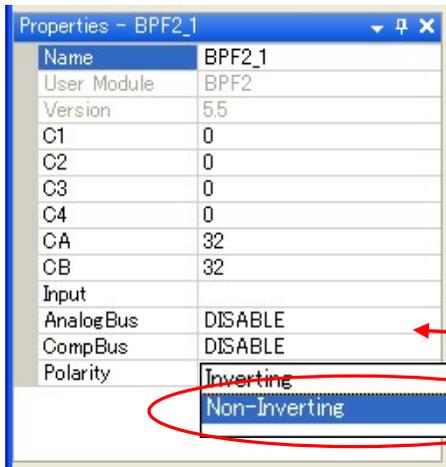
フィルター設計ウィザードの起動(旧版ソフトのGUI)

ウィザードが現れない場合は、パラメータボックスに数値を入れてみてください。

BPF2_1を右クリックしてFilter Design Wizard を起動



Wizard Windowの**パラメータボックスに設計値を入力**すると**設定値が自動計算**されます。**Apply**をクリックするとプロパティーウィンドウに値が設定されます。複雑なSCの値計算が自動的に行われます。Zoomをクリックするとグラフが拡大されます。



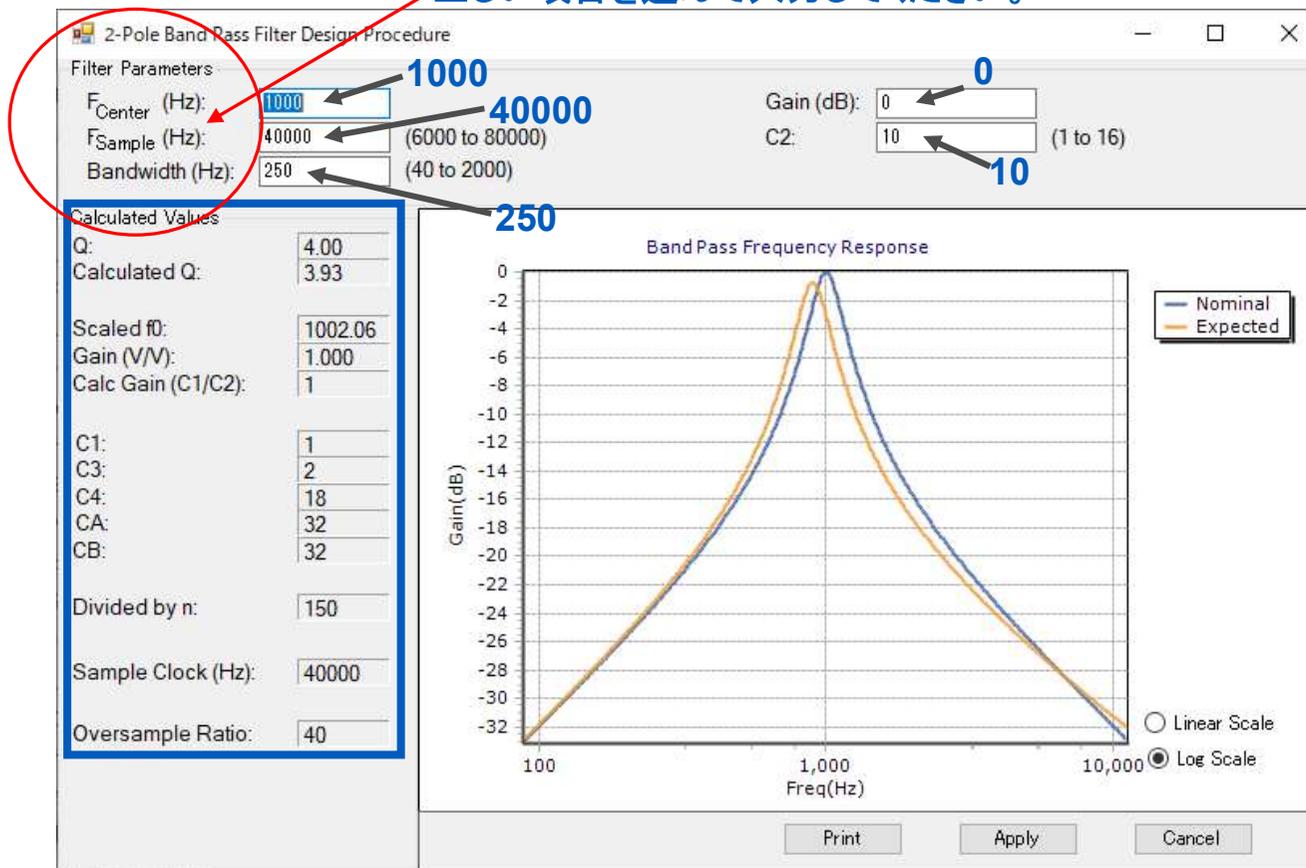
Polarity はNon-Invertingに設定

1KHz, BW250Hz, $F_{\text{Sample}}=40\text{K}$ で設計を行います(新版ソフトのGUI)

F_{Center} , F_{Sample} , Bandwidth, C2値を入力してください。

SCの設定値は自動的に計算されます。

バージョンによっては、パラメータの並びが異なる場合がありますから、正しい項目を選んで入力してください。



F_{Sample}

サンプリング周波数

を上げるほど、

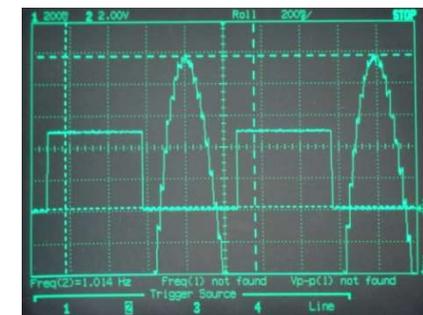
フィルタリング後の
波形はスムーズ

になります。

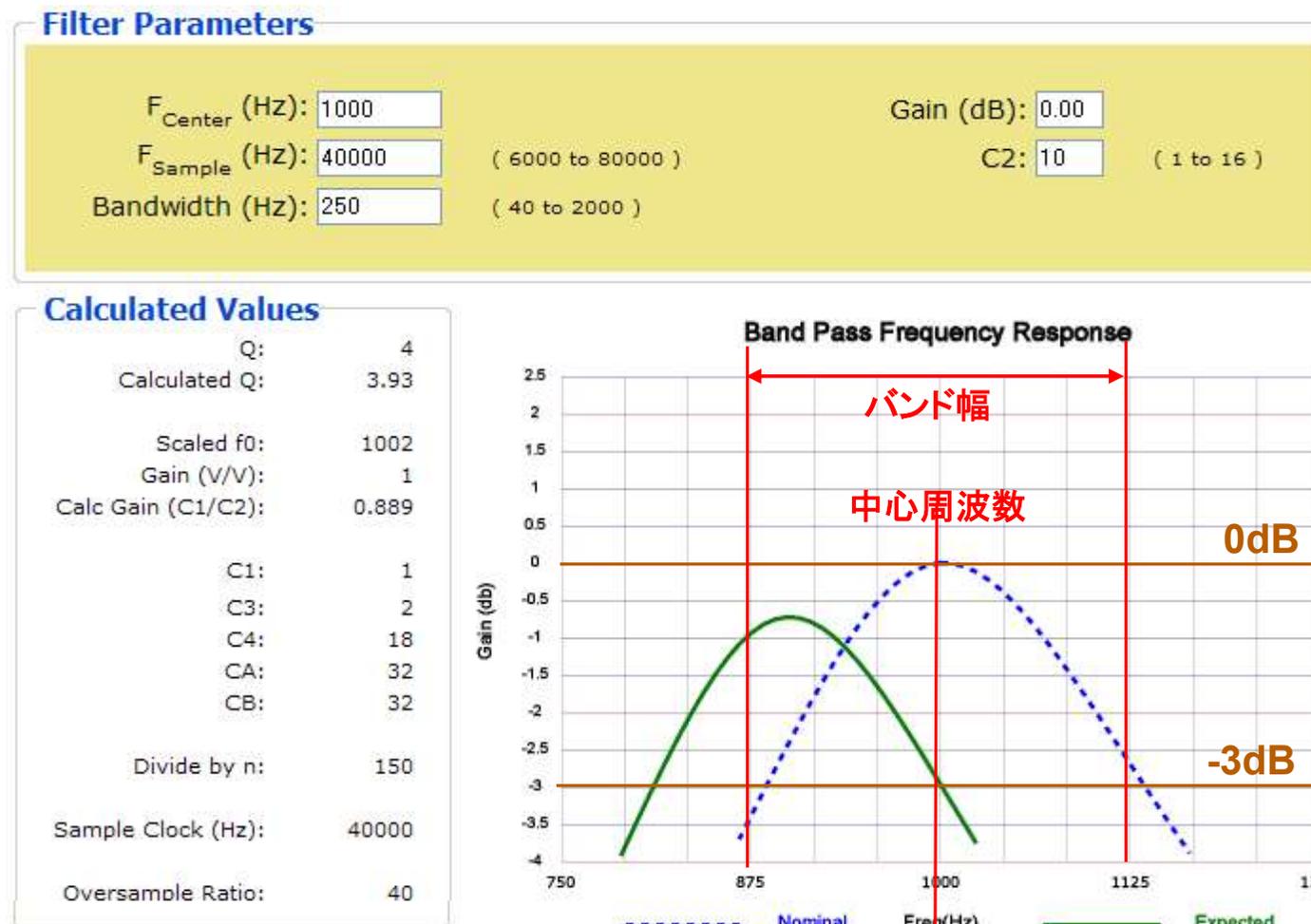
このオーバー

サンプルレートが

低いと波形には
段々がつきます。



1KHz, BW250Hz, $F_{\text{Sample}}=40\text{K}$ で設計します(旧版ソフトのGUI)

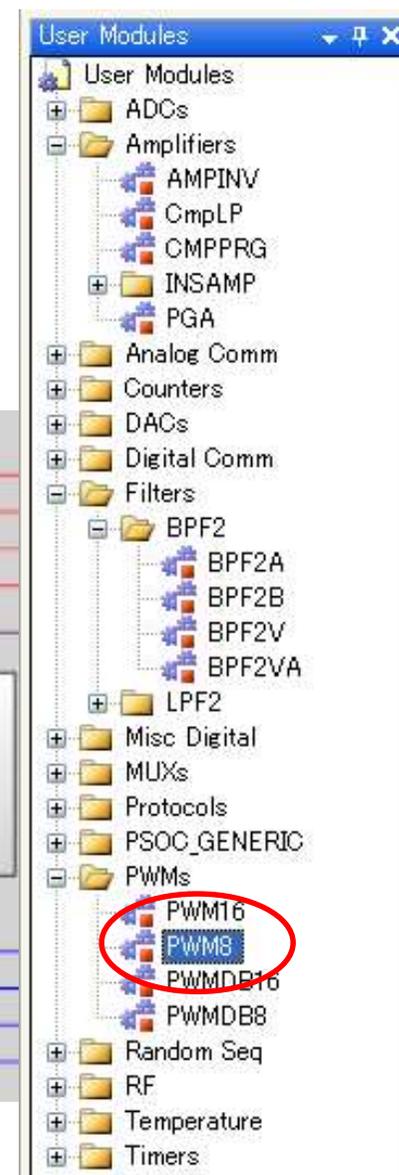
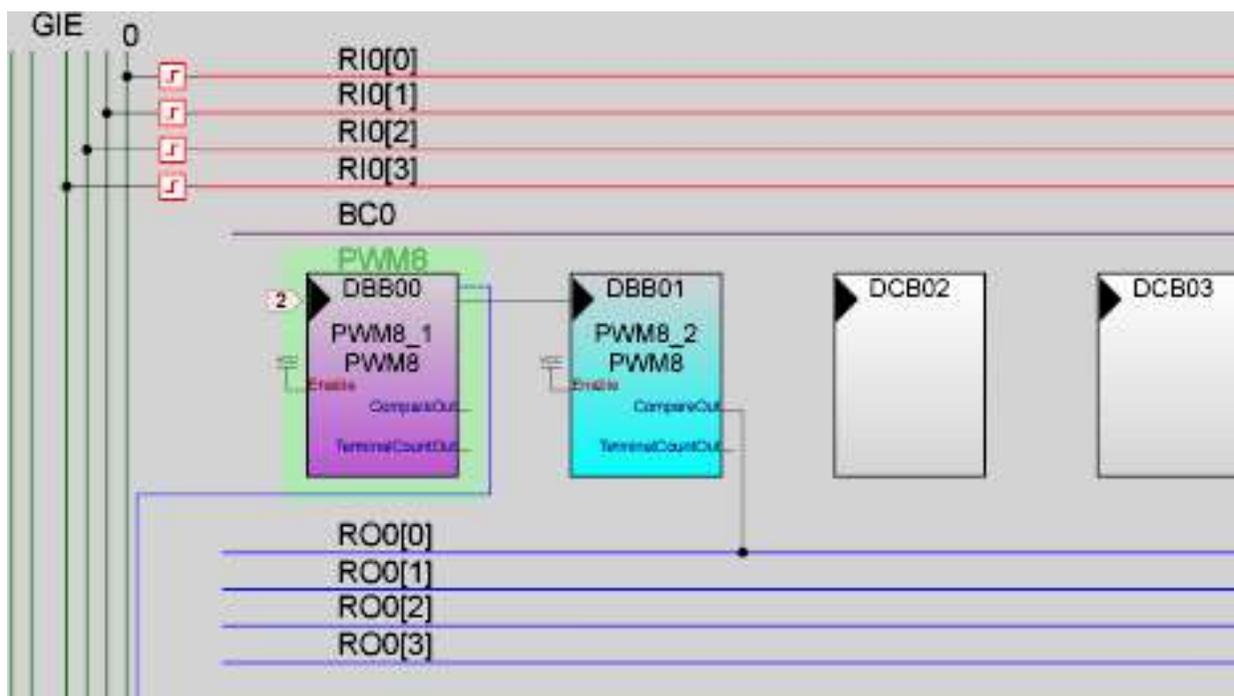


グラフ内の
青の点線は
計算値です。
緑の実線は
予測される
性能です。
予測性能と
実測値を
あとで比較
してみます

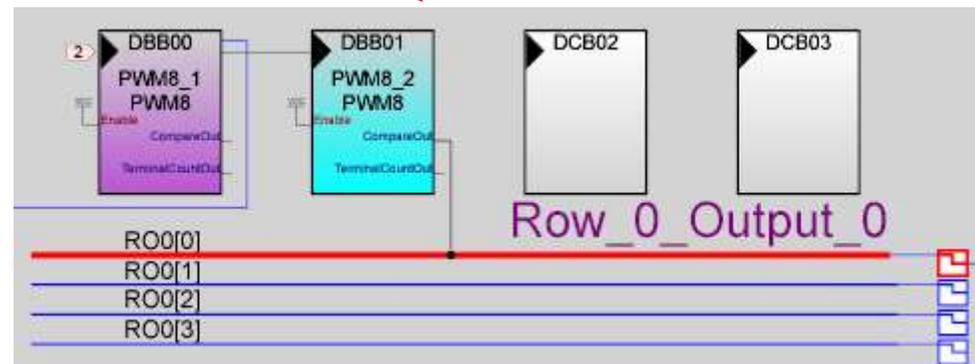
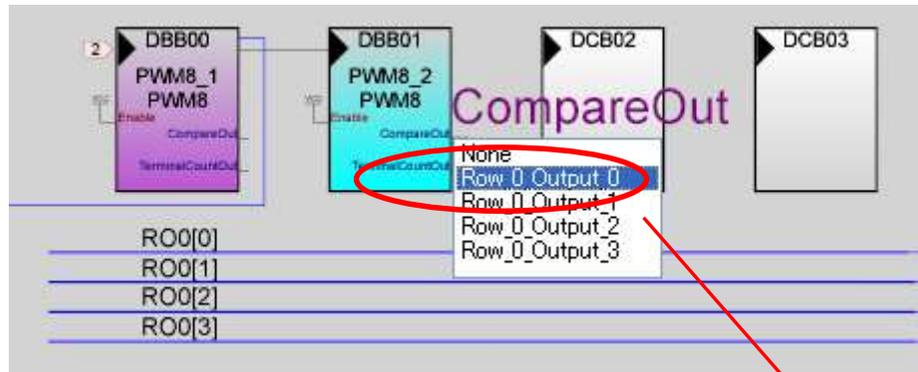
ここで $Q=1000\text{Hz}/250\text{Hz}=4$ となります。BPFモジュールに与えるクロック周波数は、 $40\text{KHz} \times 4=160\text{KHz}$ となります。このクロックはPWMで作ります。設定を確認したら **APPLY** をクリックしてPWMの設計にすすみます。160KHzを158KHzで近似しています。

PWM_1,PWM_2を配置します

PWM_1はDBB00,PWM_2はDBB01に
配置します

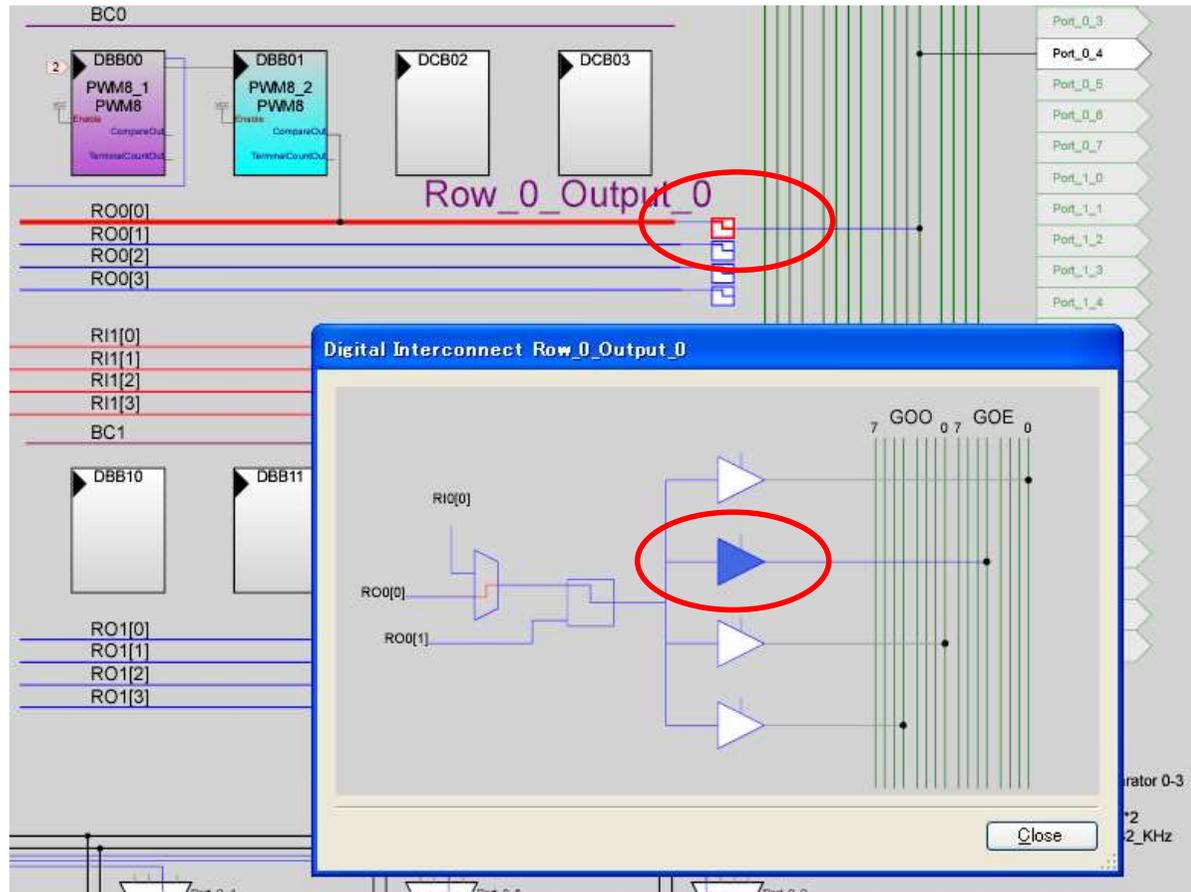


PWM_2のCompareOutをRow0_Output0に接続



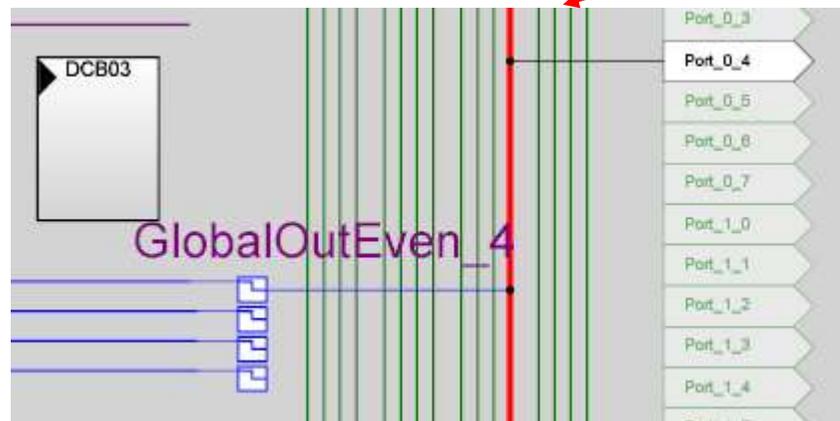
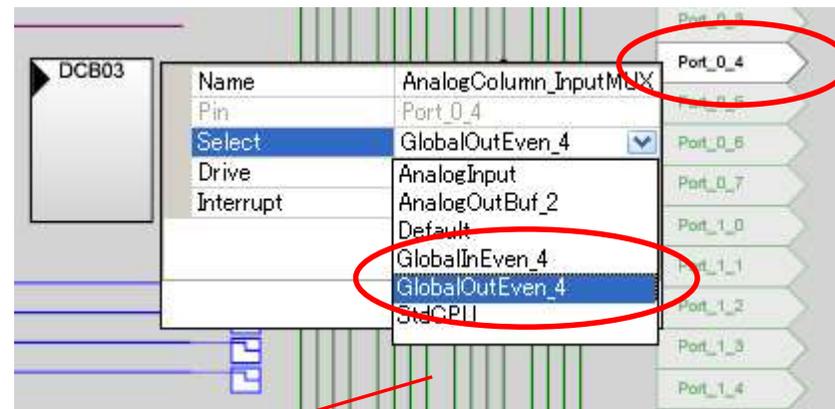
Row_0_Output_0をGOEに接続

GOE04に接続します



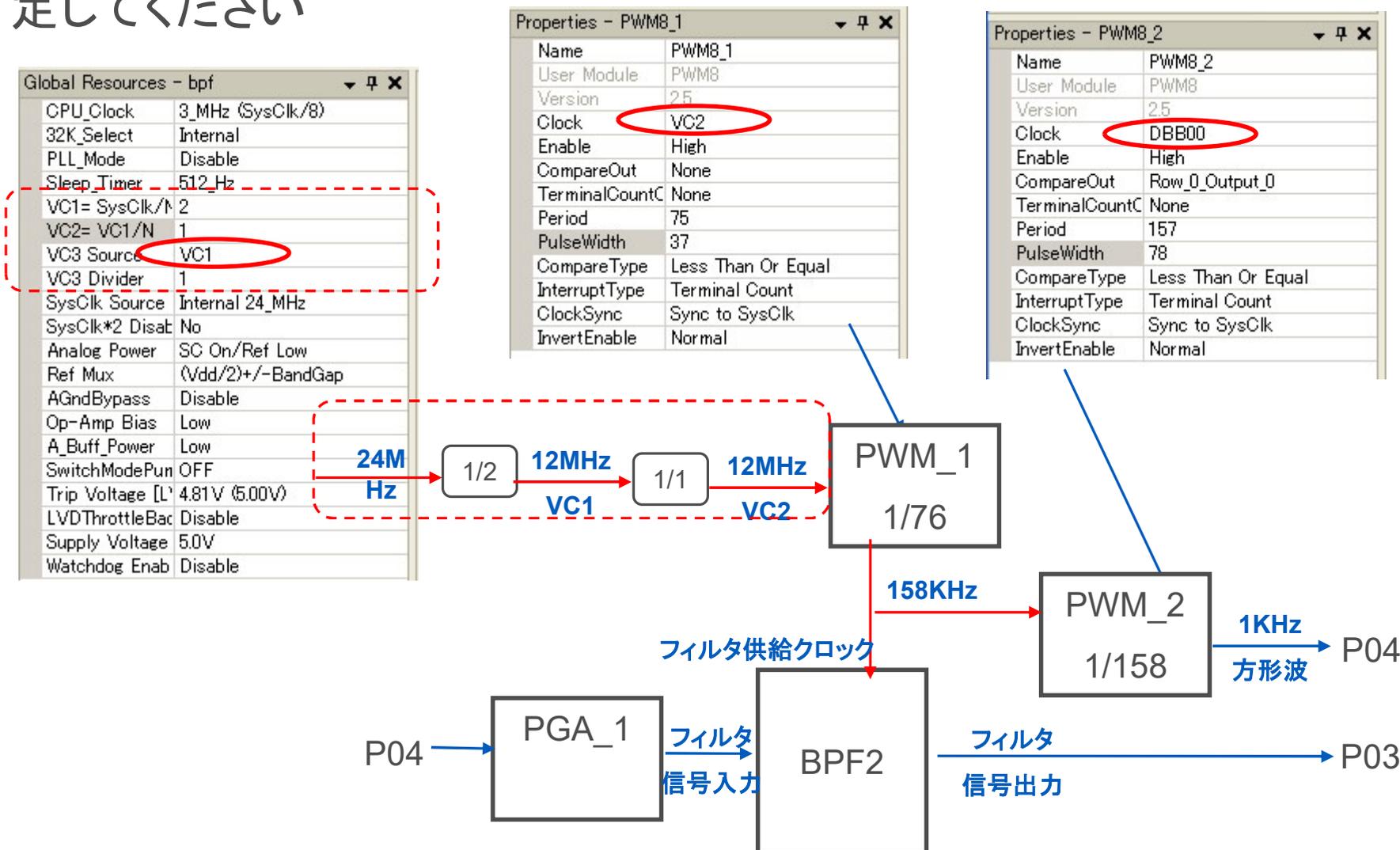
Port_04に接続

Port_04をクリックしてGOE04に接続します



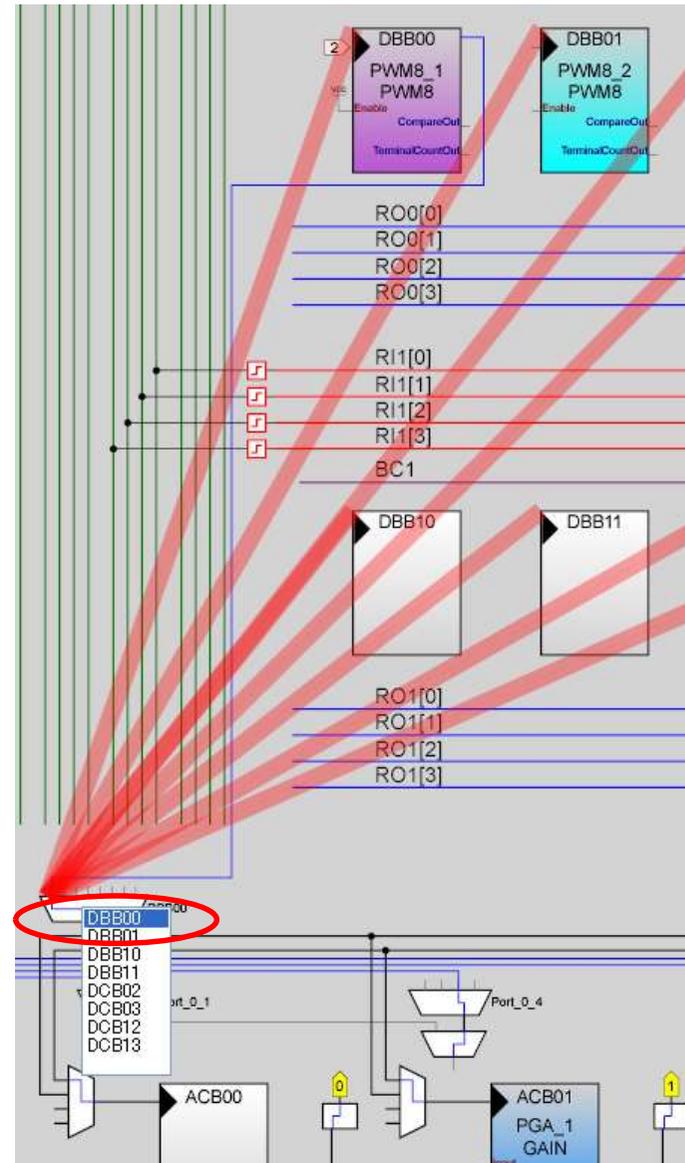
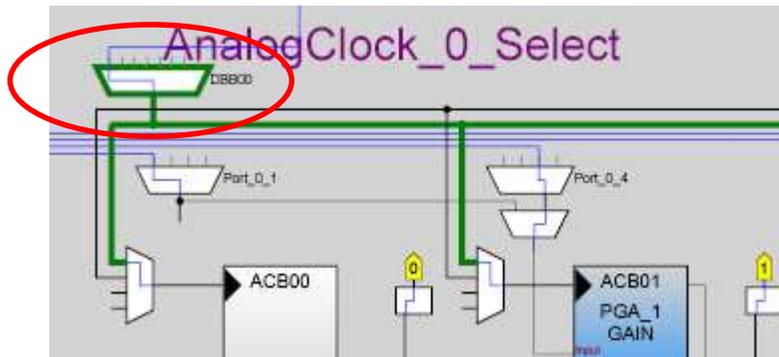
BPFに与えるクロックの作成

PWM8を2個配置してください.各UMのパラメータを以下のとおり設定してください



アナログブロックへのクロックを設定します

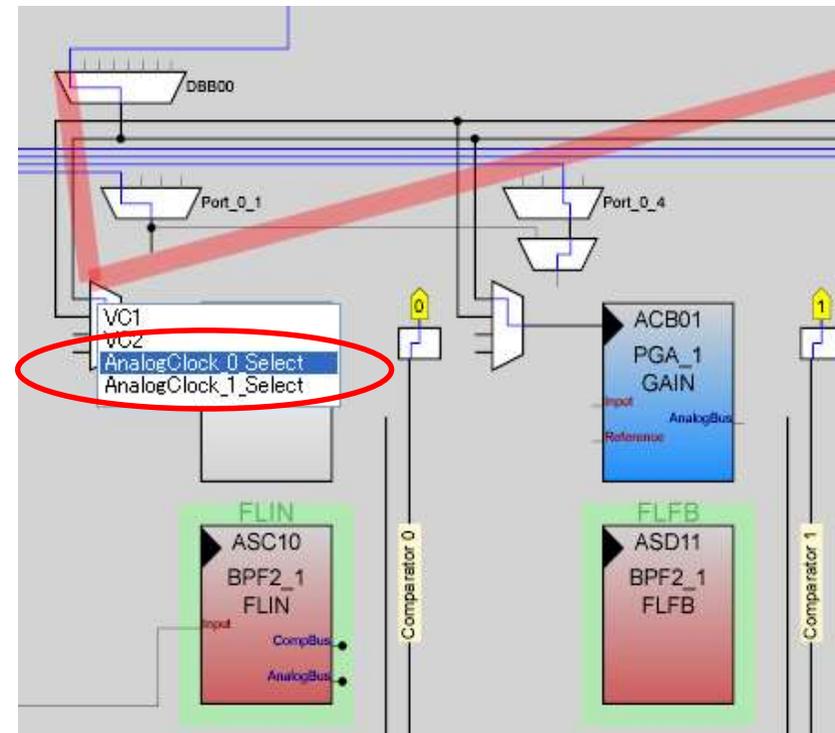
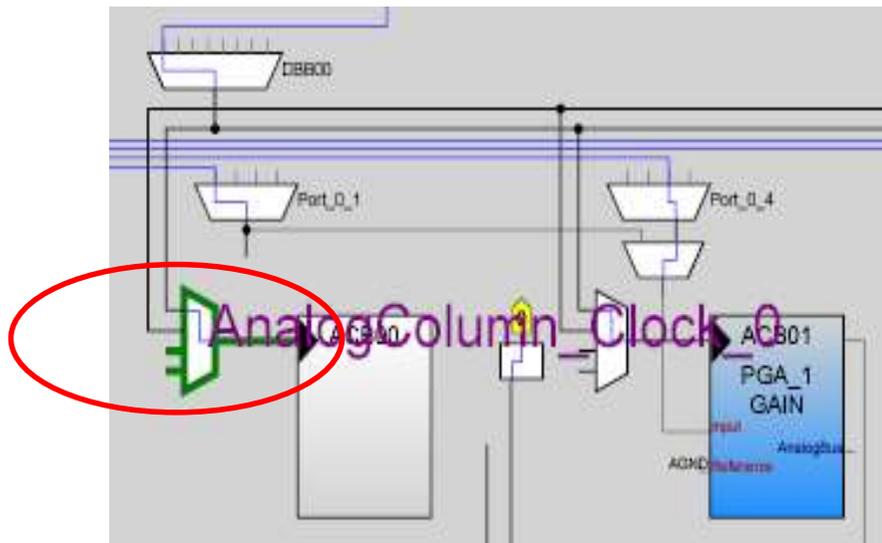
AnalogClock_0_0を左クリックしてDBB00を選択します。



左1番目の列の共通クロックを設定

AnalogColumn_Colum_Clock_0を左クリック

AnalogClock_0_Selectを選択

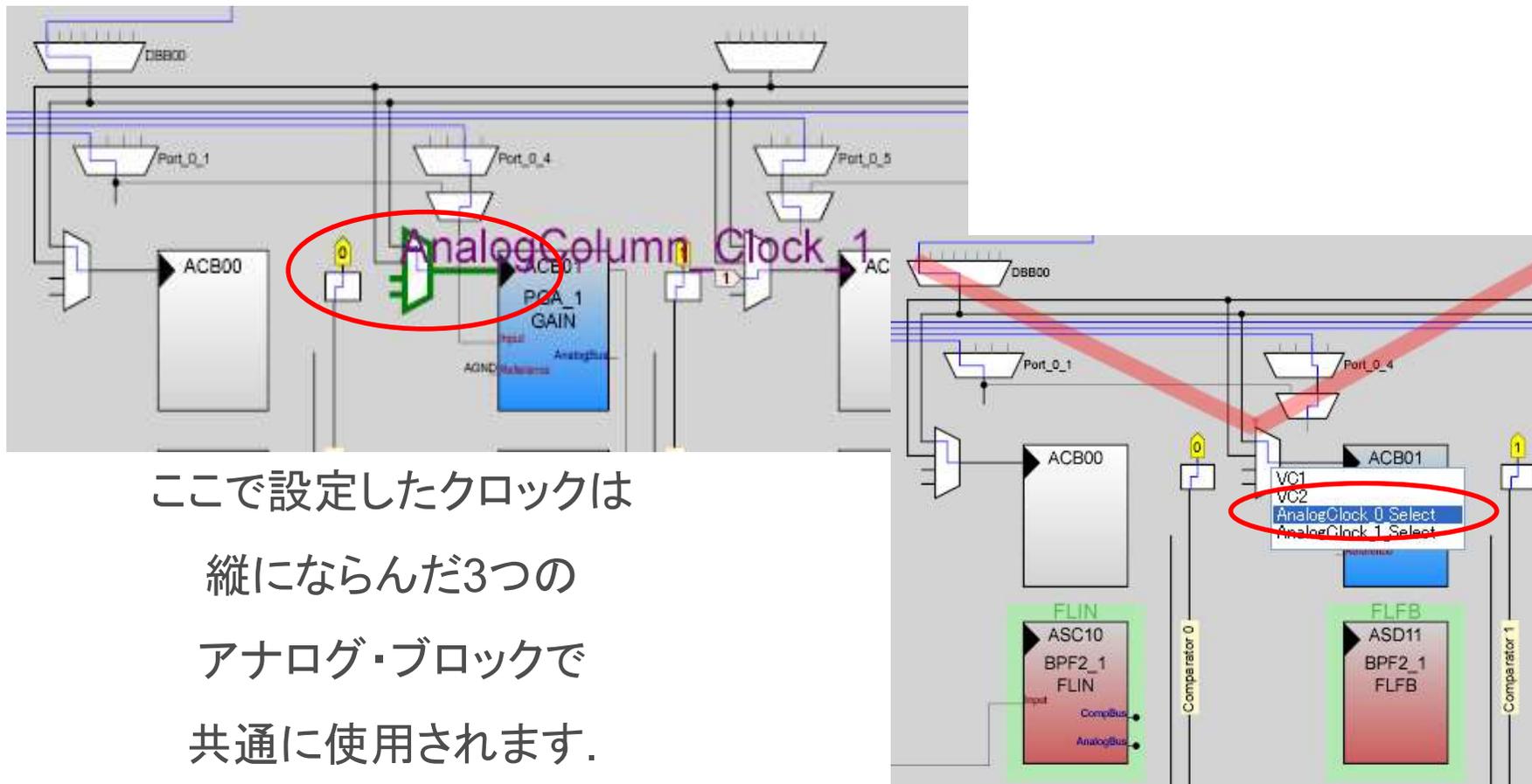


ここで設定したクロックは
縦にならんだ3つの
アナログ・ブロックで
共通に使用されます。

左から2番目の列の共通クロックを設定

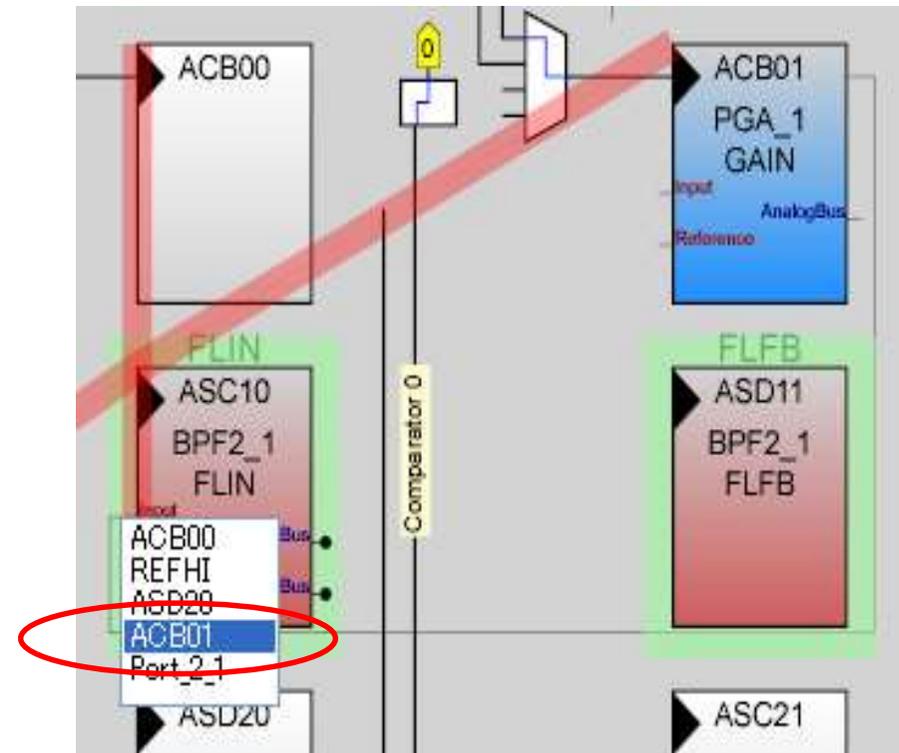
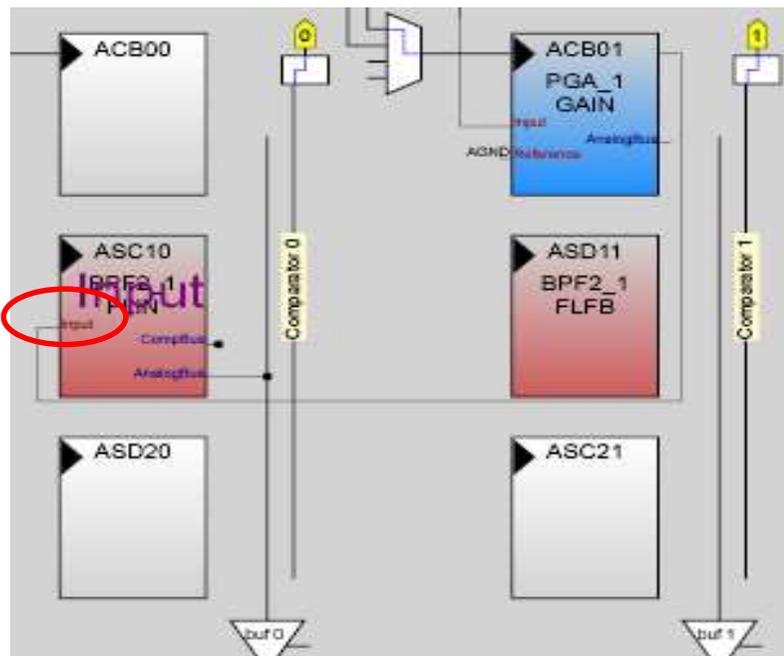
AnalogColumn_Colum_Clock_1を左クリック

AnalogClock_0_Selectを選択



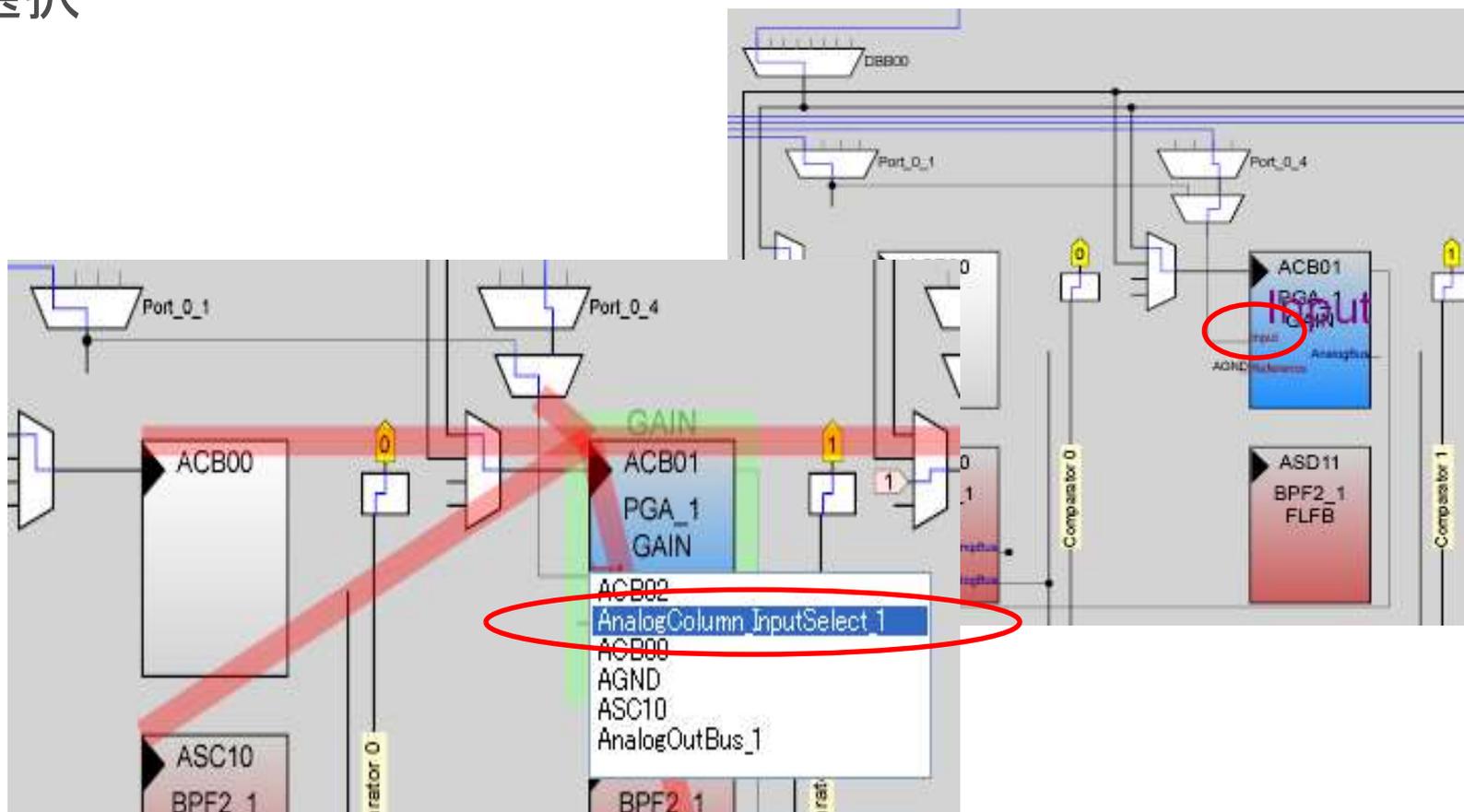
フィルタへの入力を設定

FLINの入力を左クリック,ACB01
を選択(これでPGA_1の出力が
フィルタの入口に接続されます)



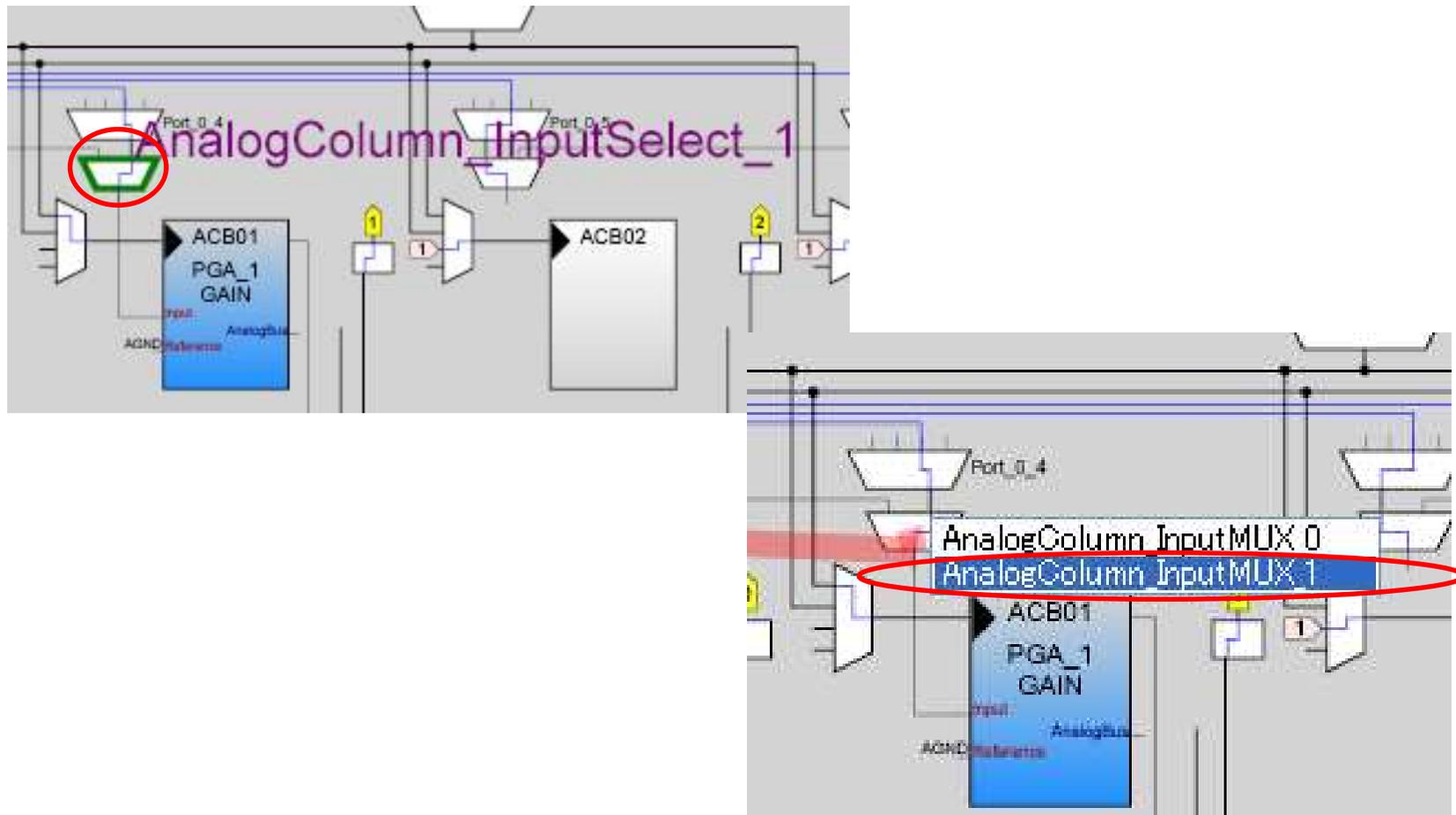
PGAへの信号入力を設定

PGA_1のinputを右クリックしてAnalogColumn_InputSelect_1を選択



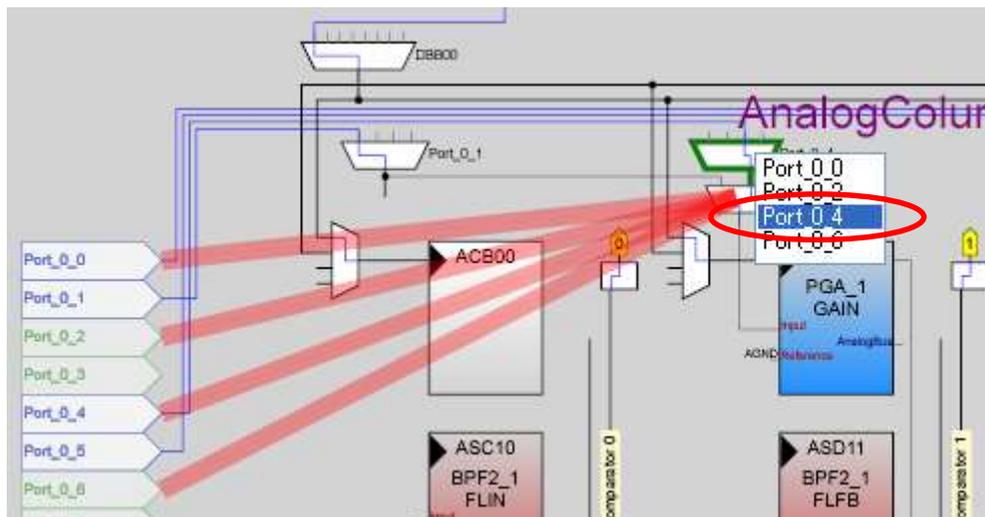
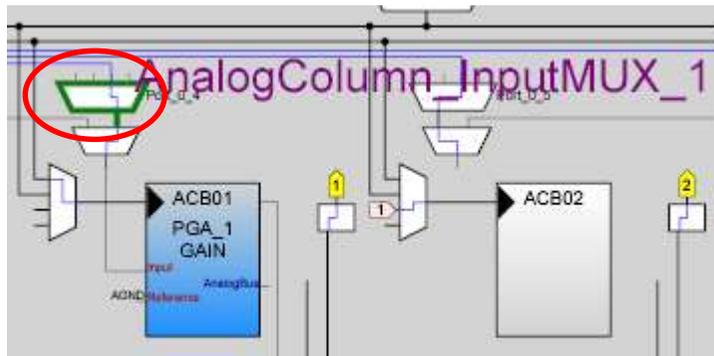
ACIS_1をACIM_1に接続

AnalogColumn_InputSelect_1を左クリックして
AnalogColumn_InputMUX_1を選択



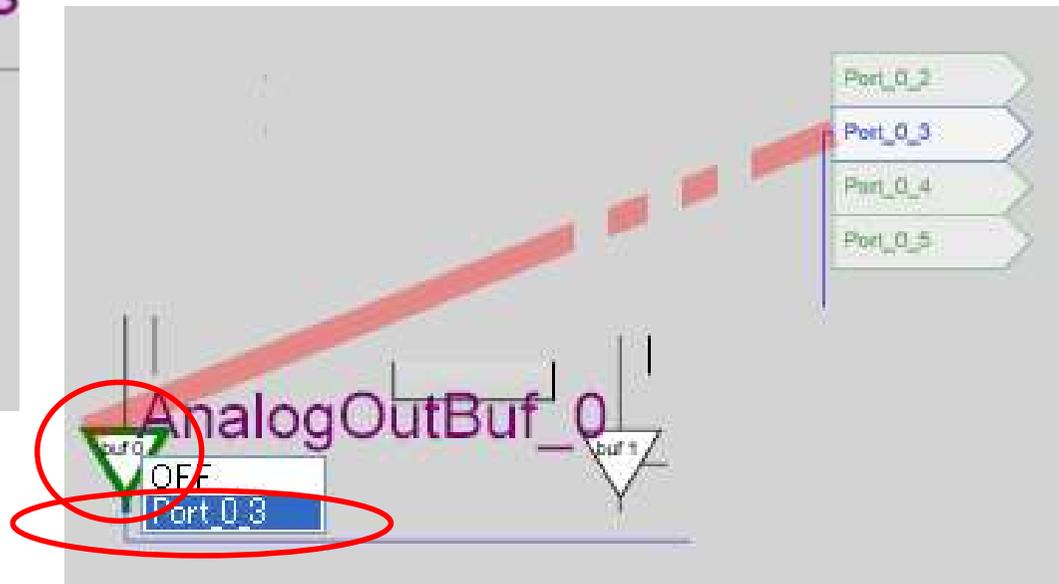
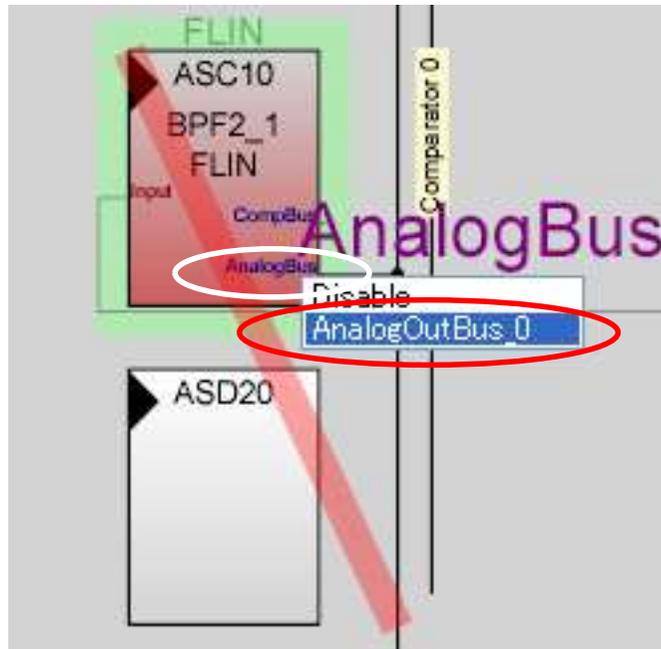
ACIM_1をPort_0_4に接続

AnalogColumn_InputMUX_1を左クリックしてPort_0_4を選択



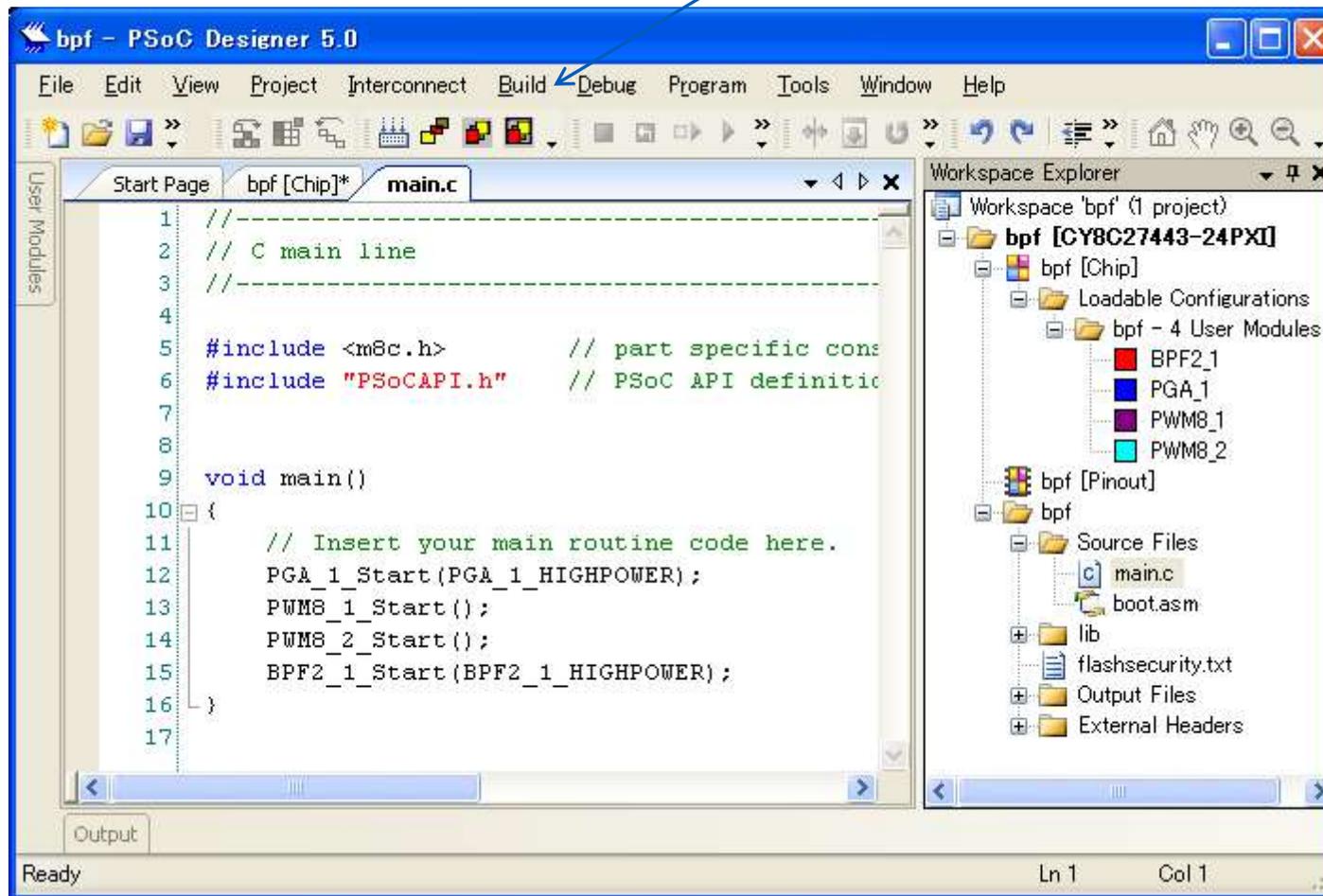
フィルタの出力をPort_0_3に接続

FLINのAnalogBusを左クリックしてAnalogBus_0を選択
AnalogBus_0を左クリックしてPort_0_3を選択



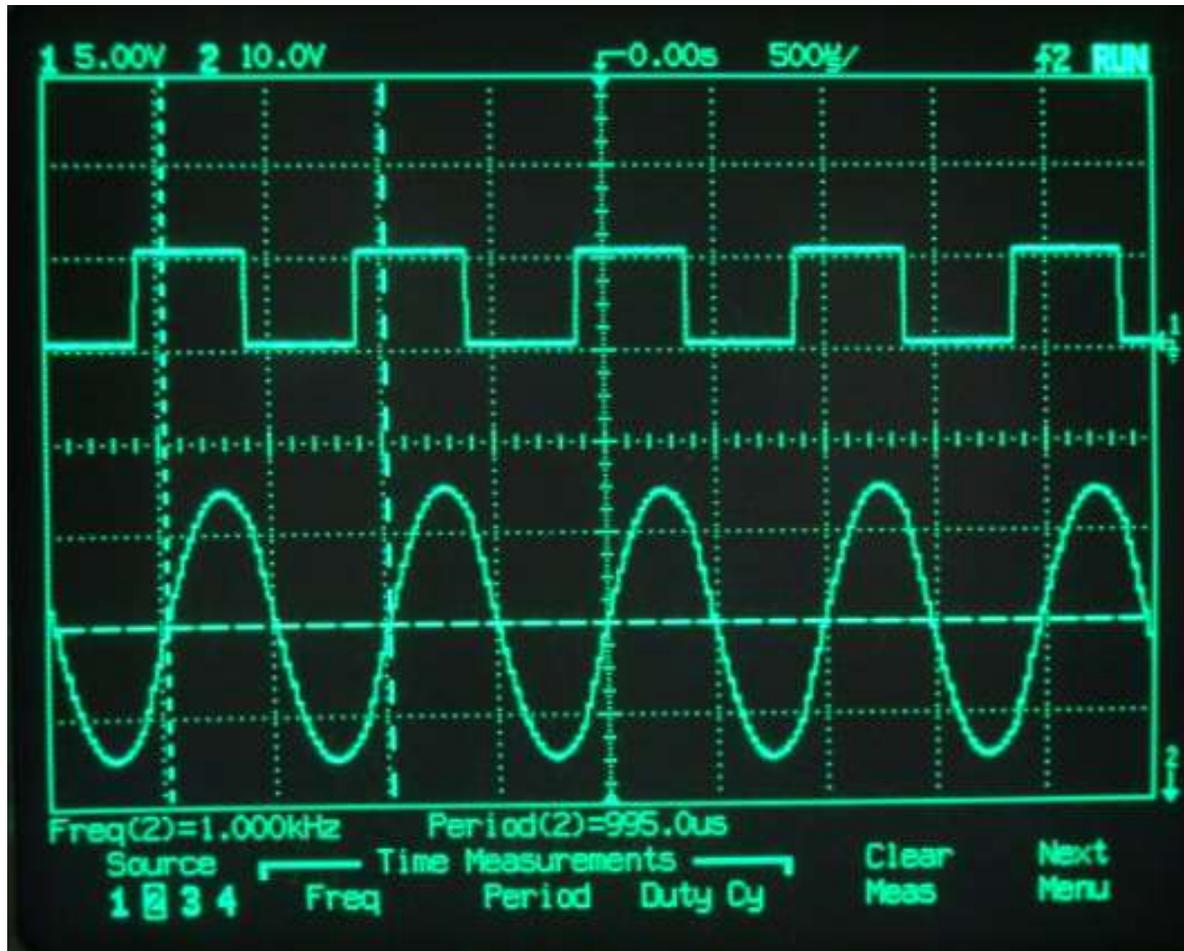
main.c のソースを記述

Build タブから、順にGenerateApplication,Compile,Build を実行してデバッグしてください。



27443にプログラムして波形観測

このように波形が観測されました。

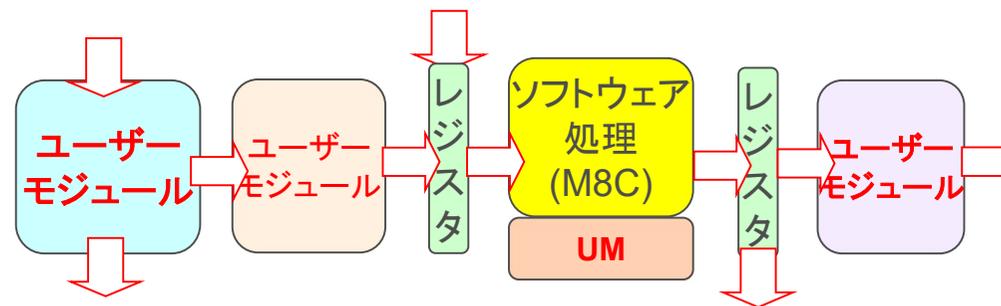


オシロスコープが
なくとも
スピーカーを
つないで、
1KHzの方形波
と正弦波の
音を聴くことが
できます。
WSでスペクトルを
見てみましょう。
WGから発生した
音と聞き比べて
みましょう



配置配線について

PSoCのパズル





モジュールの配置と相互の配線

PSoCの配線リソースは多くありません

そのため各ブロックや内部のバス、入出力ピンには、相互に配線可能なところと配線できないところがあります。

この関係を理解して配線になれる必要があります。

複数のブロックを使用するユーザーモジュールには複数の配置オプションを選択できるようになっており配線できない場合は配置パターンを変更します。

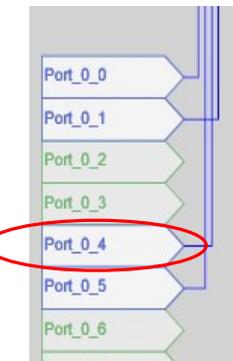
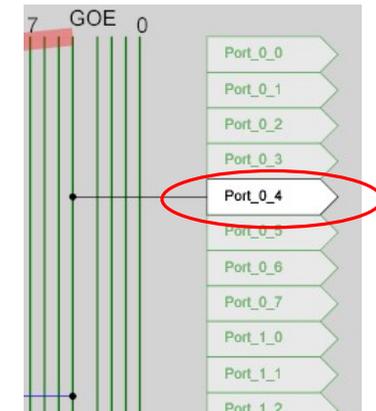
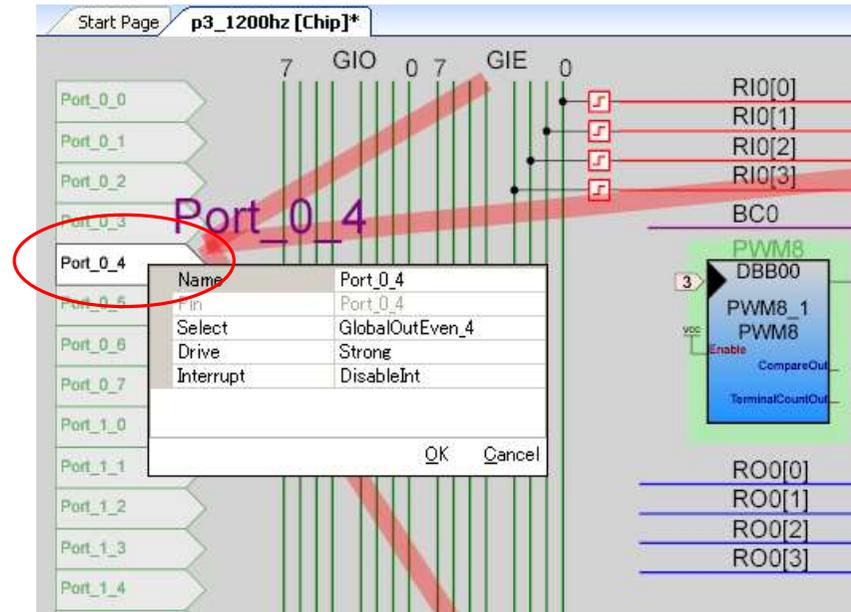


入出力ピンの例

Port番号は
上下左右の
4ヶ所に表示
されています。

上がデジタル
ブロック用,下
がアナログブ
ロック用,左
が入力で右
が出力です。

外部ピンを利用
して内部
信号を接続
することがで
きます



たとえばPort_0_4を左クリックする
とここから接続可能なガイド線が表
示されます。

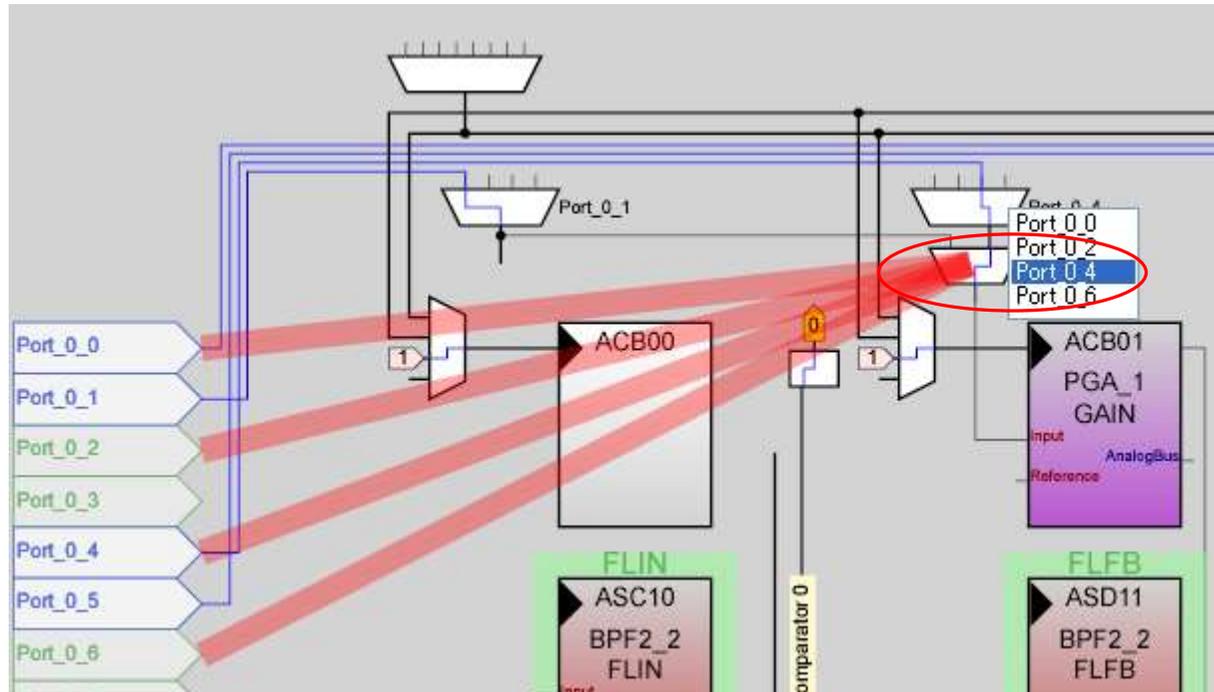
またPort_0_4は上下左右の4ヶ所
に表示されており,フレキシブルに使
えることがわかります。





Analog Column Input

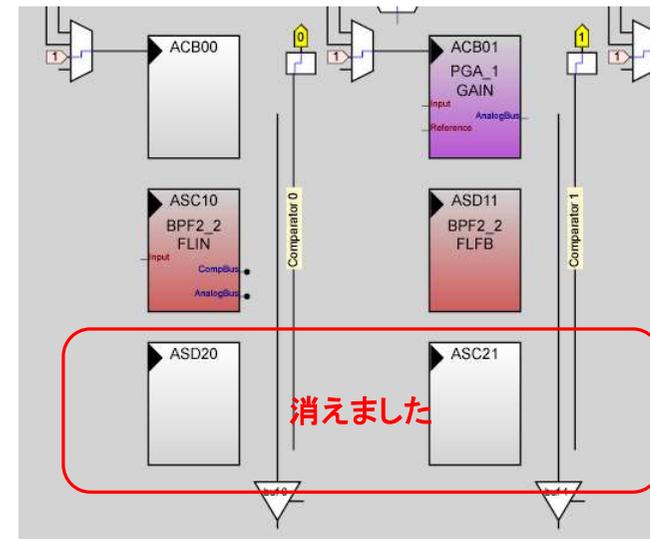
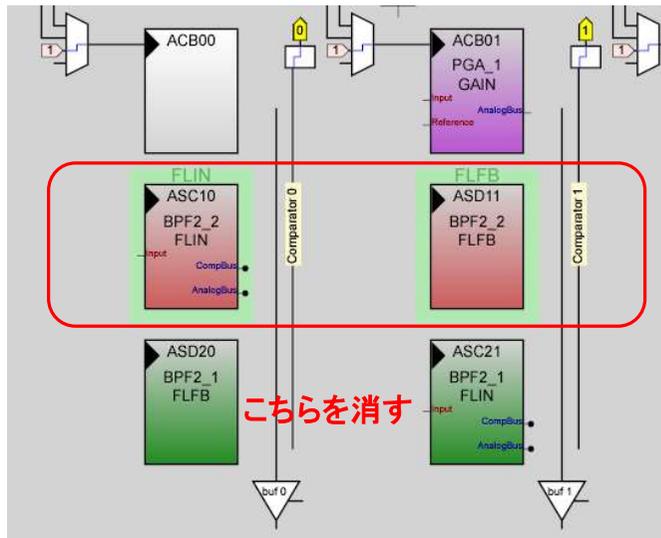
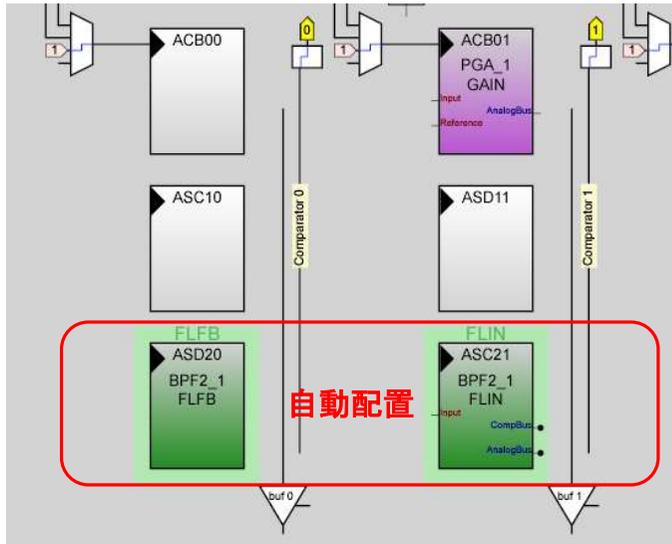
Analog Column Input Muxを右クリックすると入力信号をもってこられるPortが接続ガイドラインで表示されます. Port_0_4から信号を入力したい場合にはPGA_01の配置場所はACB01となります.





ブロックふたつの移動

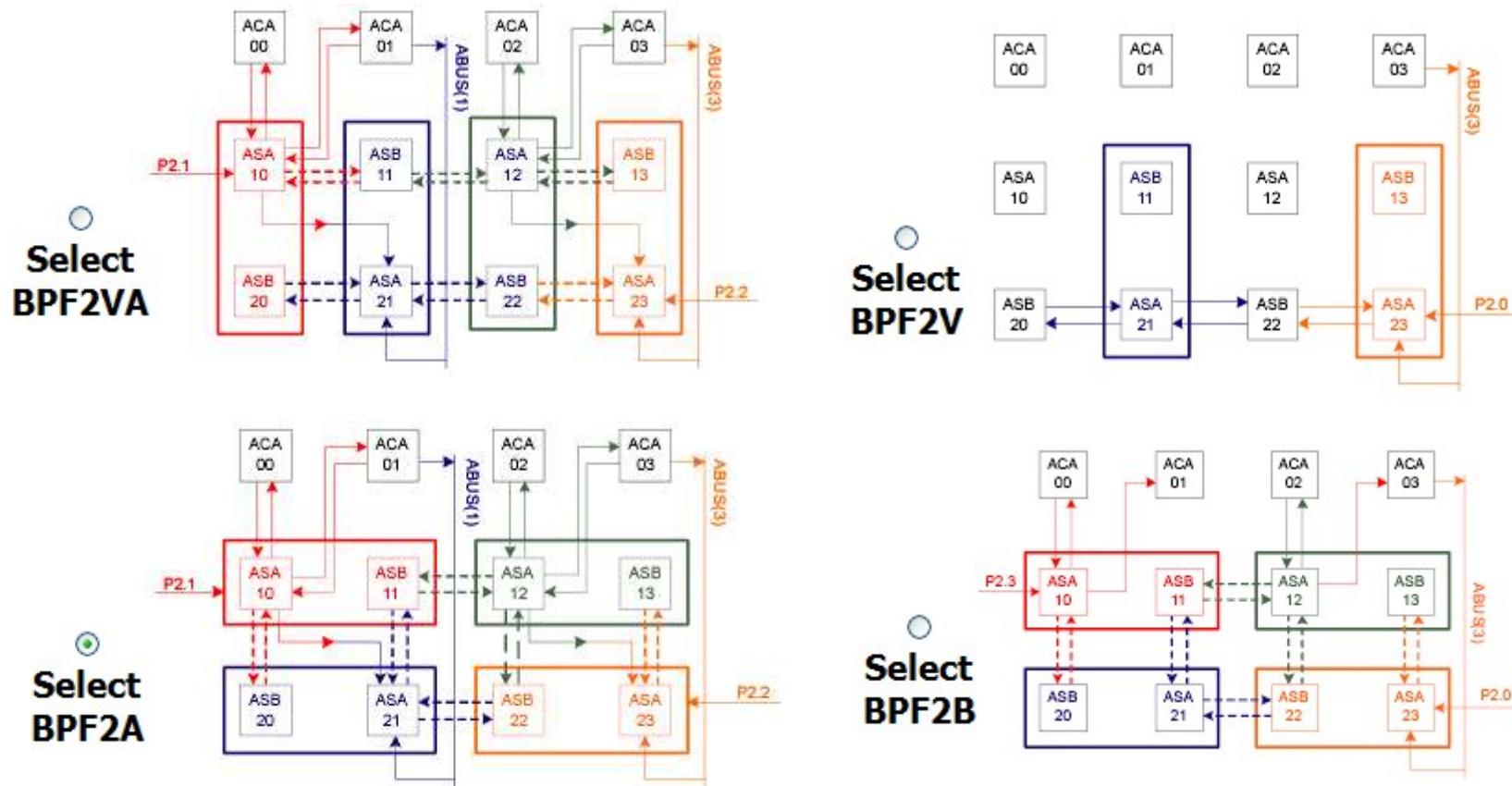
2つのブロックを使うユーザーモジュールを配置した場合中段に配置したくとも下段に配置されるときは、同じモジュールをもう一個追加配置してから下の段を削除する方法があります。





ユーザーモジュールの配置

ブロック2つを使用するBPF2の場合信号の入力先のモジュールや配線路用にタテ2段とヨコ2段の合計4とおりに配置が選択可能



Memo

フォローアップURL (Revised)

<http://mikami.a.la9.jp/meiji/MEIJI.htm>



担当講師

三上廉司(みかみれんじ)

Renji_Mikami(at_mark)nifty.com (Default - Recommended)

mikami(at_mark)meiji.ac.jp (Alternative)

http://mikami.a.la9.jp/_edu.htm

Memo

フォローアップURL

<http://mikami.a.la9.jp/meiji/MEIJI.HTM>



担当講師

三上廉司(みかみれんじ)

Renji_Mikami(at_mark)nifty.com (Default - Recommended)

mikami(at_mark)meiji.ac.jp (Alternative)

http://mikami.a.la9.jp/_edu.htm