

インテル® FPGA FIR II IP コア 簡易チュートリアル

Ver.22.1

目次

本書をお読みにする前に.....3

1. はじめに4

2. 使用環境4

3. 係数データの構築.....4

4. 係数データの取り込み..... 16

5. フィルター形態の指定..... 20

6. 入出力仕様の設定 23

7. 出力ビット幅の切り出しと端数処理 24

8. 回路規模および実装関連の設定 25

9. まとめ..... 30

改版履歴 31

本書をお読みにする前に

この資料の内容は 2023 年 12 月現在のものです。

この資料は、2. 使用環境 を条件に作成しています。それ以外のバージョンやデバイスファミリーでも共通に適用できる内容もありますが、一部適さない場合もありますので ご注意ください。

文書中の記号

① Note	補足情報などを記載しています。
② Point	重要なポイントを記載しています。
📖 参考	理解を深めるため、参考となる資料やサイトを紹介しています。
⚠️ 注記	この資料の中では具体的には触れませんが、必要となる知識や情報を記載しています。
🚫 禁止	注意点や、してはいけないことを記載しています。

文中の表記

<u>下線</u>	クリックする事で、資料中の別の章や、外部のサイトにジャンプします。
太字斜体	画面の操作をする際の、メニューやウィンドウなどに表示されている文字を示しています。
xxxxxxx	入力するコマンド文字列を示しています。

1. はじめに

この資料は、インテル® 社の FPGA に実装可能な FIR フィルター向け IP コアであるインテル® FPGA FIR II IP コア(以降、FIR II) を使ったシンプルなデザインを構築するための簡単な操作手順を説明するための簡易チュートリアルです。係数データの生成フローについては、MathWorks® 社の MATLAB Online で利用可能なフィルター解析ツールを使用する方法を紹介しています。

2. 使用環境

この資料では、以下の開発ツールを使用しています。

表 1. この資料に必要な開発環境

No.	環境	概要
1	インテル® Quartus® Prime Standard Edition (以降、Quartus® Prime)	インテル® FPGA のハードウェアを開発するためのツールです。 この資料では、インテル® Quartus® Prime Standard Edition 22.1 を使用しています。

3. 係数データの構築

最初に、デジタル・フィルターの設計で不可欠なフィルター係数を、MathWorks® 社開発ツール MATLAB を使用して構築します。

① Note:

MATLAB は、数値計算やデータ処理など多くのアプリケーションに使用されるツールであり、信号処理や、その他の幅広いエンジニアリングや科学的分野で利用されています。大学・研究機関や産業界においても広く使われています。

この資料では、MATLAB の実行に、MATLAB Online を利用します。以下の Web サイトにアクセスして、**[MATLAB Online の使用を開始する]** ボタンをクリックします。

<https://jp.mathworks.com/products/matlab-online.html>

上記 Web サイトにアクセスすると、執筆時点では図 3-1 のような画面が表示されます。



【図 3-1】 MATLAB Online ①

① Note:

MATLAB Online は、Web ブラウザから MATLAB を実行することができるオンライン版の MATLAB です。インストールやライセンスの管理などが不要で、簡単に MATLAB を使うことができます。MATLAB Online は、無償版と有償版があり、ブラウザとインターネットに接続できる場所であれば、いつでもどこでも MATLAB を利用することができます。

① Note:

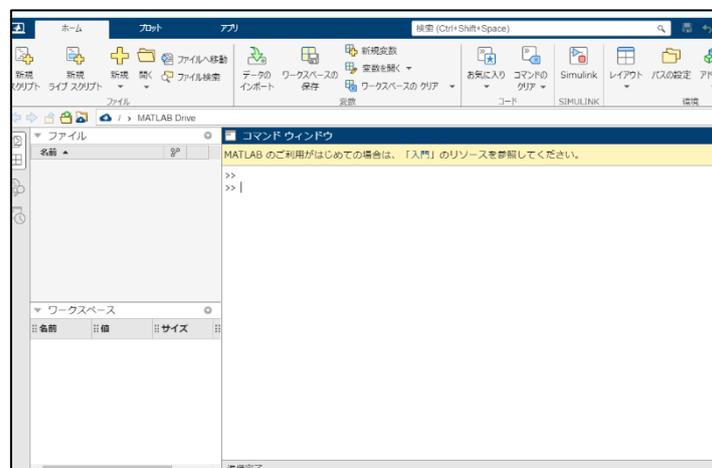
MATLAB Online の無償版は、執筆時点では、ユーザー登録後、月 20 時間以内で利用できます。

図 3-1 の [MATLAB Online の使用を開始する] ボタンをクリックすると、図 3-2 の画面が表示されたら、[MATLAB Online (basic) を開く] ボタンをクリックして、MATLAB を起動します。



【図 3-2】 MATLAB Online ②

図 3-3 は、MATLAB 起動時のメイン画面です。



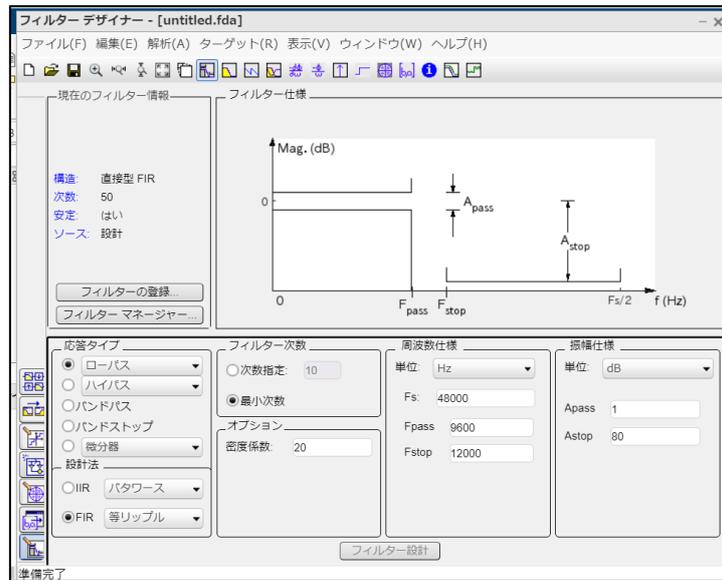
【図 3-3】 MATLAB Online メイン画面

MATLAB 起動後、フィルター解析ツールを起動するためにプロンプトから `filterDesigner` と入力します。



【図 3-4】 filterDesigner の起動

filterDesigner が起動して、図 3-5 のような画面が表示されます。



【図 3-5】 filterDesigner 起動時の画面

以降、このフィルター解析ツールに対して、以下の仕様を設定します。

フィルター設計仕様

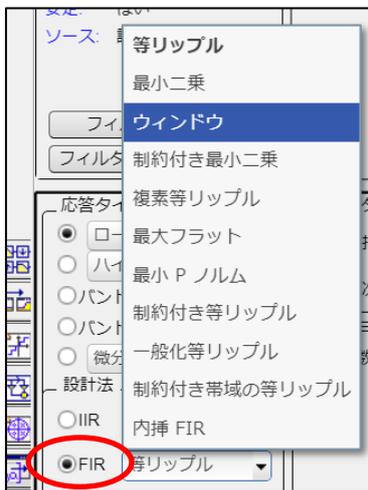
- ・ フィルターの形態: FIR フィルター
- ・ フィルターの種類: ローパス・フィルター
- ・ カットオフ周波数 (fc) : 8kHz
- ・ サンプリング周波数 (fs): 100 MHz (= 100000 kHz)
- ・ 窓関数: ハニング窓(ハン窓)
- ・ タップ数: 37

最初に、**応答タイプ**を **ローパス** に指定します。



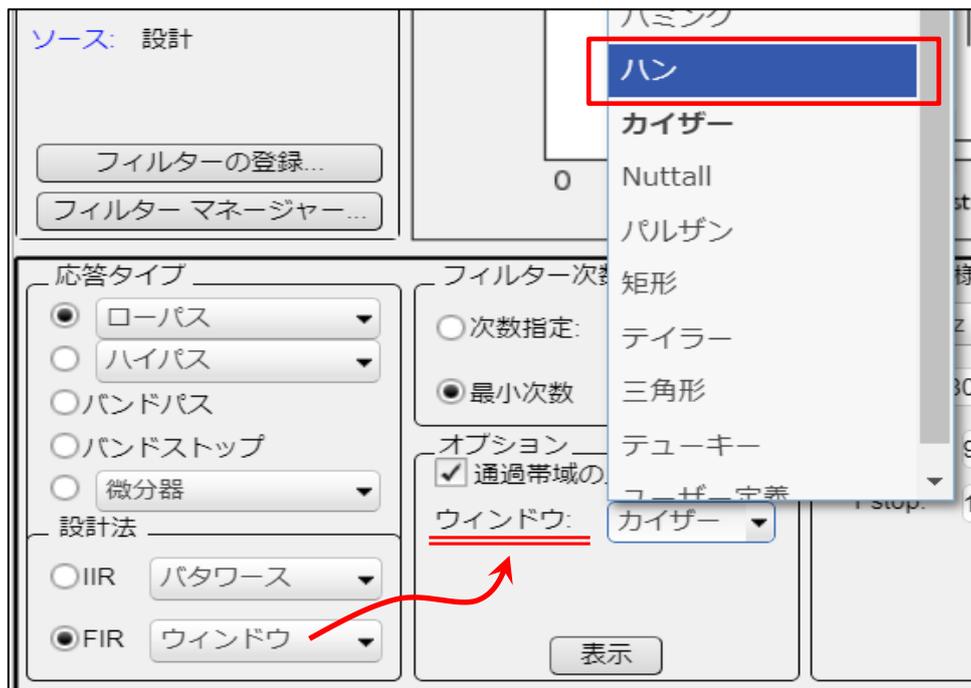
【図 3-6】 filterDesigner ~ 応答タイプの指定 ~

FIR にチェックを入れて、プルダウン・メニューより **ウィンドウ** を指定します。



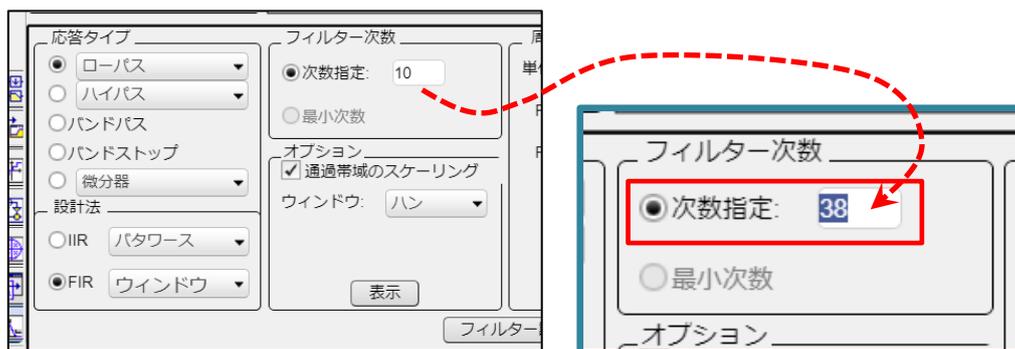
【図 3-7】 filterDesigner ~ ウィンドウの指定 ~

ウィンドウ を指定した後、新たに **オプション** 欄が画面内に表示されます。
ウィンドウ のプルダウン・メニューより、**ハン** を選択します。



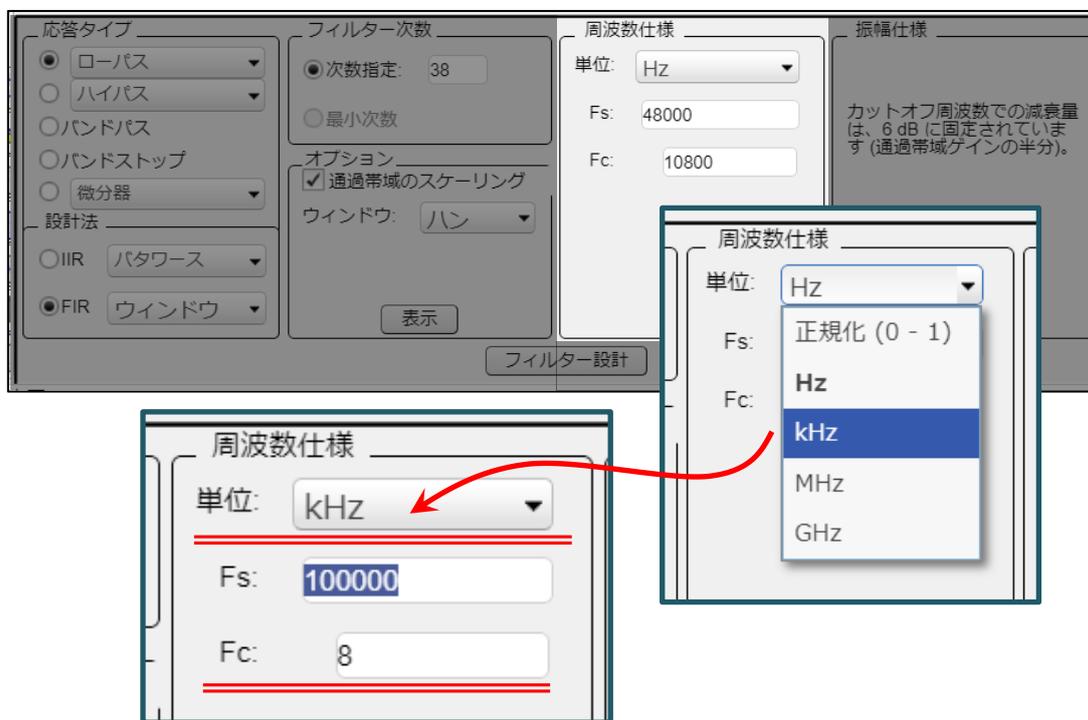
【図 3-8】 filterDesigner ~ ウィンドウ および オプション の指定 ~

フィルター次数 欄の **次数指定** に **38** を指定します。



【図 3-9】 filterDesigner ~ フィルター次数 の指定 ~

周波数仕様 欄では、単位: をプルダウン・メニューから kHz を選択し、Fs: 欄に 100000 を入力し、Fc: 欄には 8 を入力します。



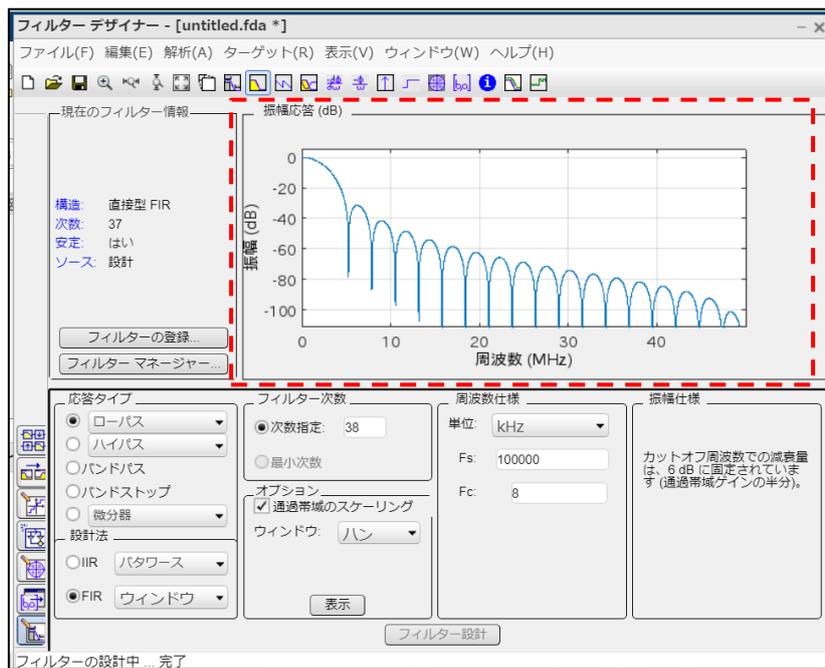
【図 3-10】 filterDesigner ~ サンプル周波数とカットオフ周波数の指定 ~

画面最下段の [フィルタ設計] ボタンをクリックします



【図 3-11】 filterDesigner ~ フィルタ設計の実行 ~

これまで設定した内容は、画面内の **振幅応答 (dB)** 欄に特性図として反映されます。



【図 3-12】 filterDesigner ～ 周波数応答特性 ～

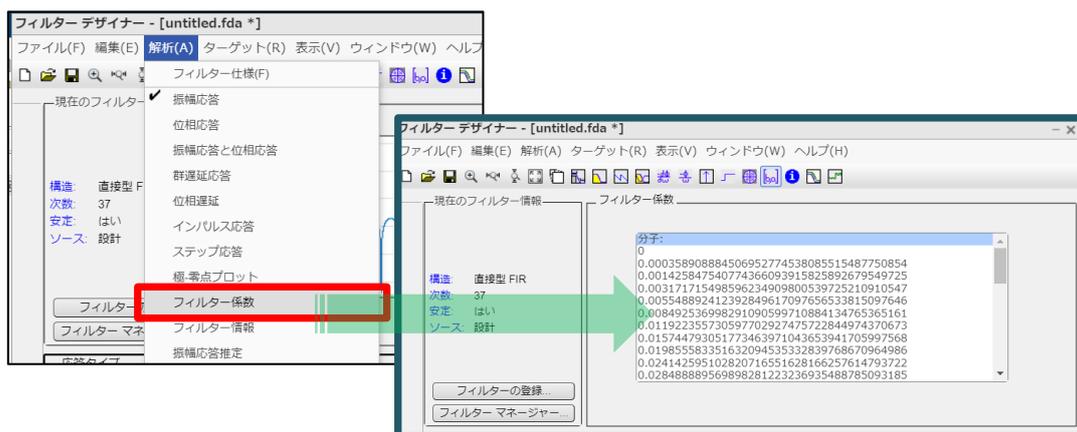
注記:

今回指定した窓関数では、対称性を考慮して、最終的に **次数** は、37 に換算されます。



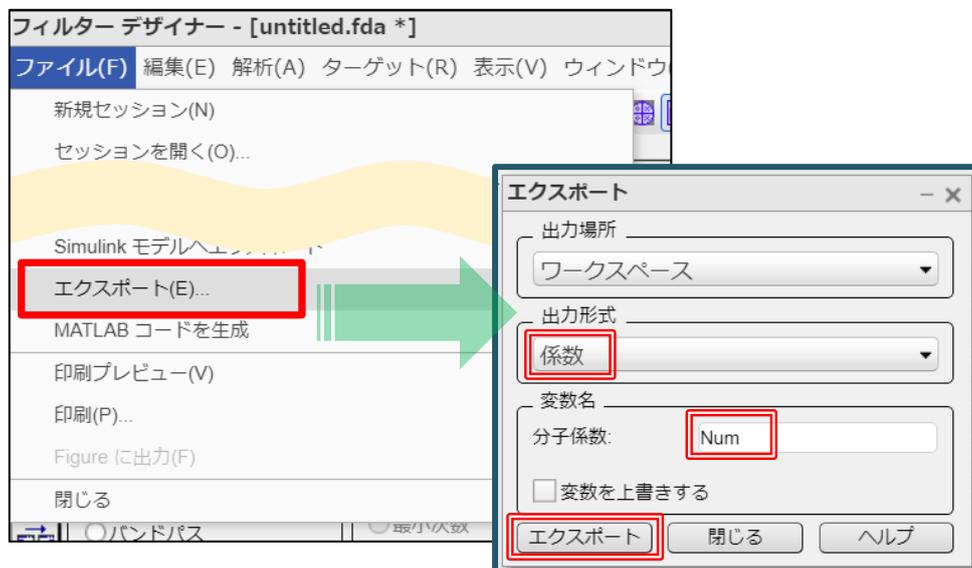
【図 3-13】 filterDesigner ～ 現在のフィルター情報 ～

解析(A) メニューから **フィルター係数** を選択すると、ツールが算出したフィルター係数が **フィルター係数** 欄に表示されます。



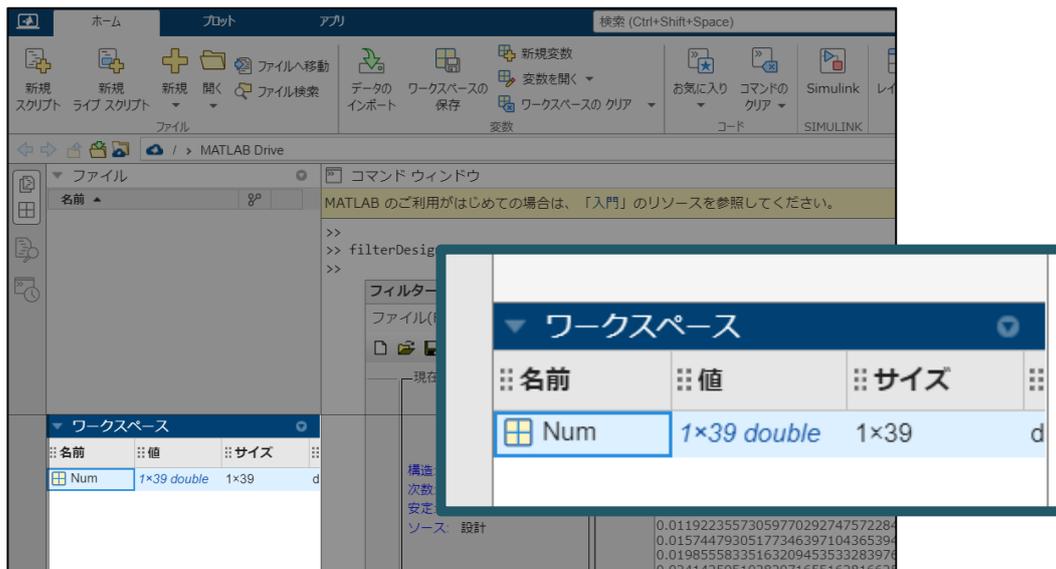
【図 3-14】 filterDesigner ～ フィルター係数の生成 ～

ファイル(F) メニューから **エクスポート(E)** を選択すると、**エクスポート** 画面が起動します。
エクスポート 画面にて、**出力形式** を **係数** に選択し、**変数名** 欄に任意の変数名を指定します。
(この資料では、デフォルトの **Num** を適用します。)
[エクスポート] ボタンをクリックします。



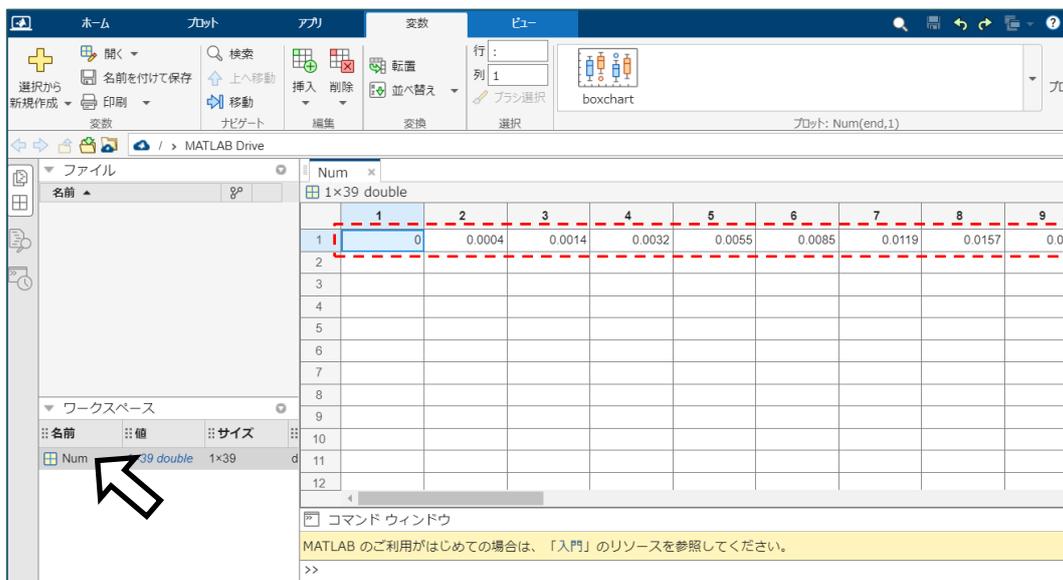
【図 3-15】 filterDesigner ～ フィルター係数のエクスポート ① ～

MATLAB のメイン画面の **ワークスペース** 欄に変数名 **Num** が表示されます。



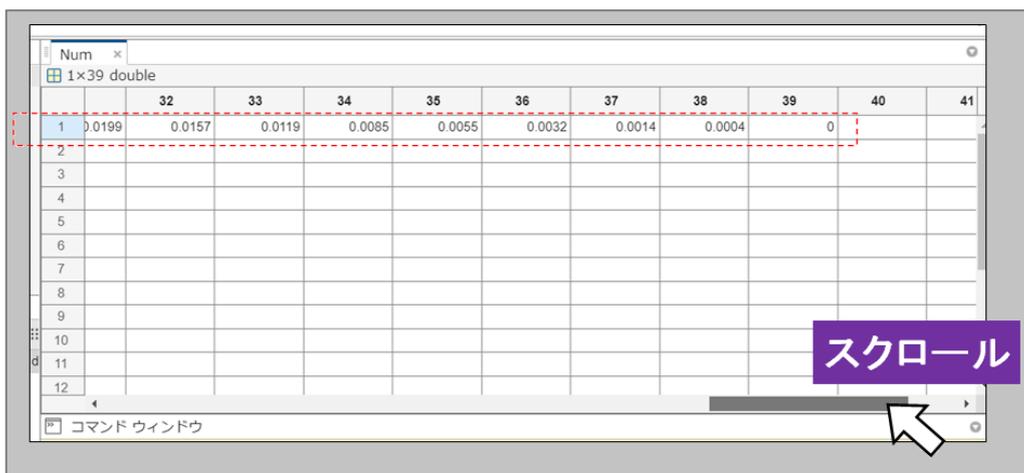
【図 3-16】 filterDesigner ~ フィルター係数のエクスポート ② ~

変数名 **Num** をハイライト後、ダブル・クリックすると、**変数** タブに切り替わります。表計算形式でフィルター係数が右方向にリストアップされていることが確認できます。



【図 3-17】 filterDesigner ~ フィルター係数のエクスポート ③ ~

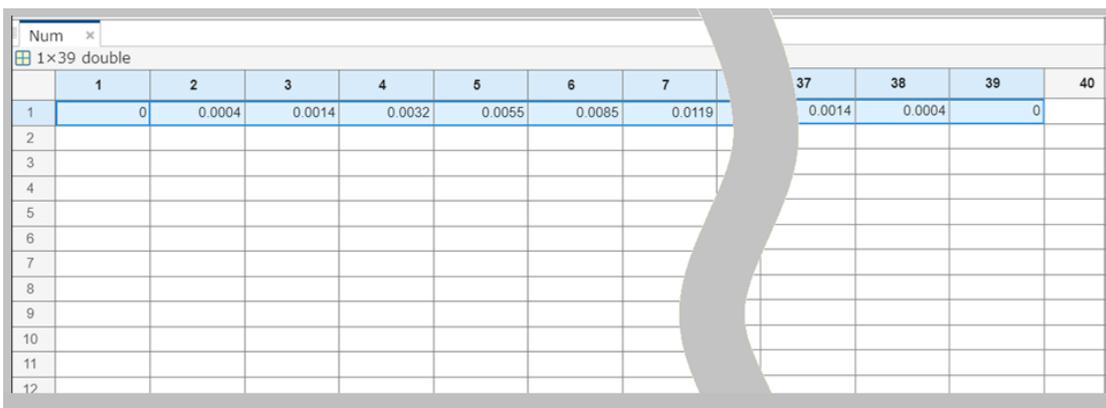
右にスクロールして、最後の係数を確認します。最後の係数は空白セルの手前にあります。



【図 3-18】 filterDesigner ～ フィルター係数のエクスポート ④ ～

以下の図に示されているように、最初の係数から最後の係数までをハイライトします。

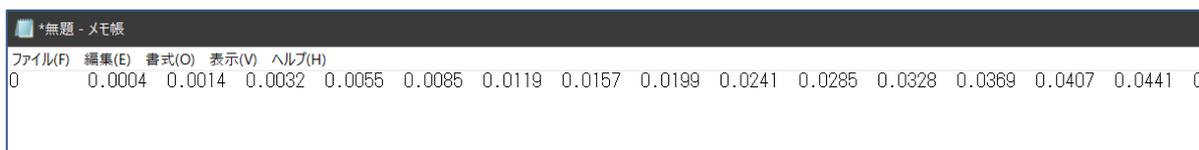
[Ctl] + [C] をタイプして、ハイライトした係数をコピーします。



【図 3-19】 filterDesigner ～ フィルター係数のエクスポート ⑤ ～

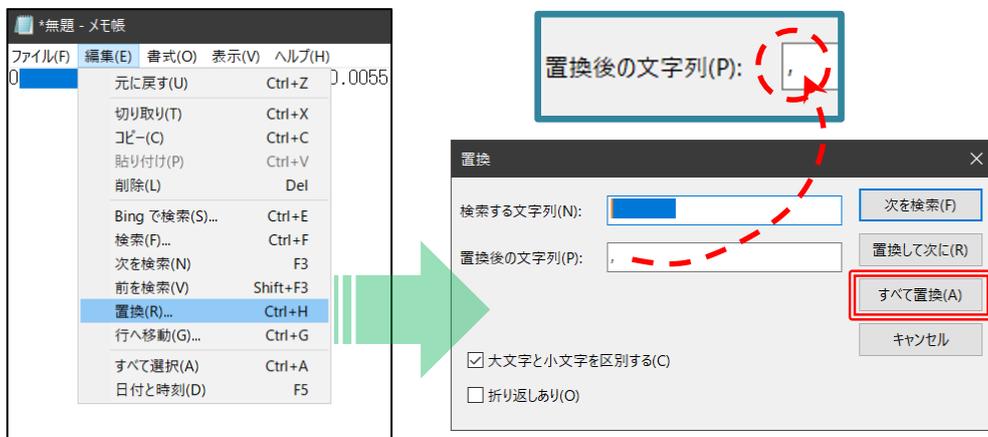
テキストエディターを開いて、[Ctl] + [V] をタイプして、テキストエディター上にペーストします。

(この資料では、テキストエディターは、Windows のメモ帳を使用しています。)



【図 3-20】 filterDesigner ～ フィルター係数のエクスポート ⑥ ～

メモ帳の場合、データ間をハイライトして [Ctl] キーと [C] キーをタイプした後、この部分をカンマ (,) に一括置換します。係数データ間が全てカンマ (,) で区切られていることを確認できたなら、係数データ用テキストファイルとして保存します。



【図 3-21】 filterDesigner ~ フィルター係数のエクスポート ⑦ ~

置換前



【図 3-22】 フィルター係数 ~ メモ帳への貼り付け ~

置換後



【図 3-23】 フィルター係数 ~ カンマ区切り (CSV) ~

係数データ間が全てカンマ(,)で区切られていることを確認できたら、係数データ用テキストファイルとして保存します。

この資料では、ファイル名 `coeff_fir_8khz_100Mhz_matlaab_fda_gen.txt` として保存します。

④ Point:

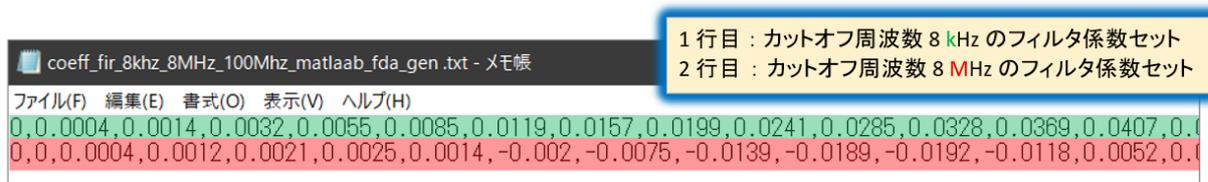
FIR II では、係数データをカンマで区切ったテキストファイルは、係数データとして認識されます。

⚠ 注記:

テキストファイルでは、1 つのフィルター仕様に対して、係数データ・セットを 1 行で記載します。

⚠ 注記:

別のフィルター仕様を追加したい場合、図 3-24 のように、次の行数に係数データ・セットを追加します。



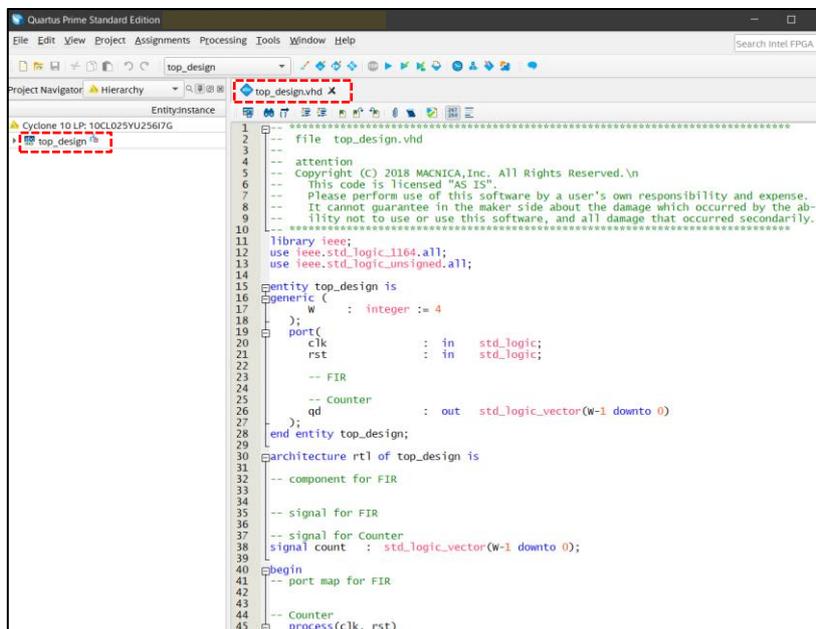
【図 3-24】複数の係数セットを追加した例

④ Note:

カンマで区切ったテキストファイルは、Excel 表計算ソフトウェアなどを使用した場合、CSV (カンマ区切り) として、拡張子 `.csv` ファイルで保存することもできます。

4. 係数データの取り込み

Quartus® Prime のプロジェクトを構築します。この資料では、プロジェクト名 *top_design* に対して、最上位階層のデザインを作成して、プロジェクト名と同じファイル名 *top_design.vhd* を適用します。

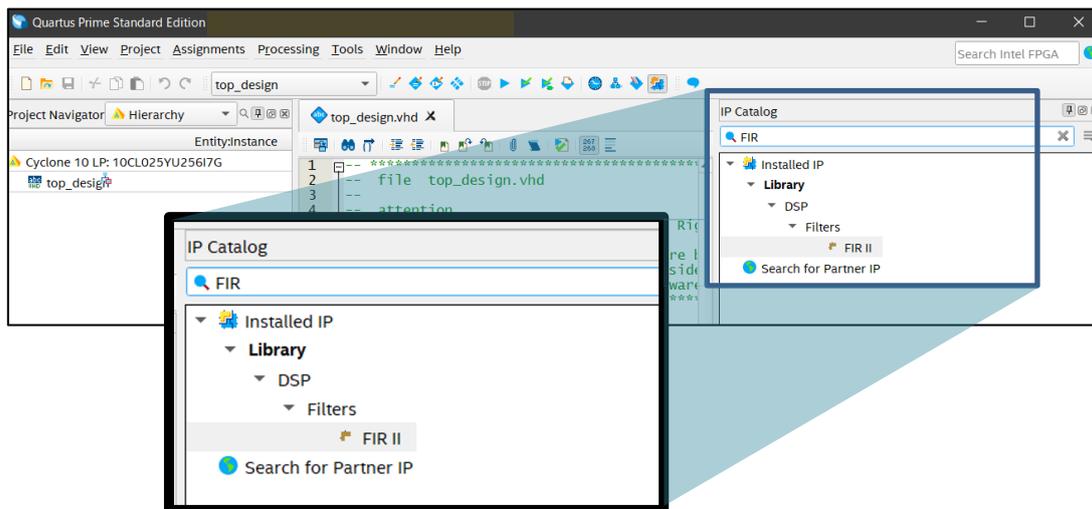


【図 4-1】最上位階層

Tools メニューから、**IP Catalog** を選択すると、**IP Catalog** 欄が表示されます。

IP Catalog 欄に **FIR** とタイプすると、ツールが該当する IP コアを検索して、**FIR II** という IP コア名が、以下のフォルダ内に表記されます。

Library – DSP – Filters

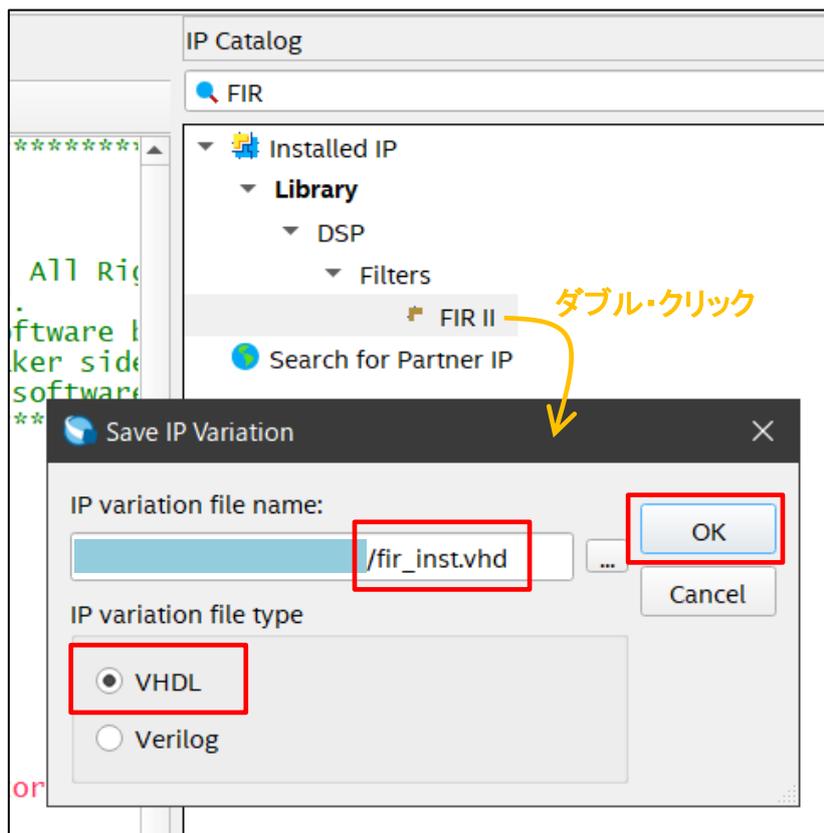


【図 4-2】IP Catalog 検索結果

FIR II をハイライトした後、ダブル・クリックすると、**Save IP Variation** 画面が起動します。

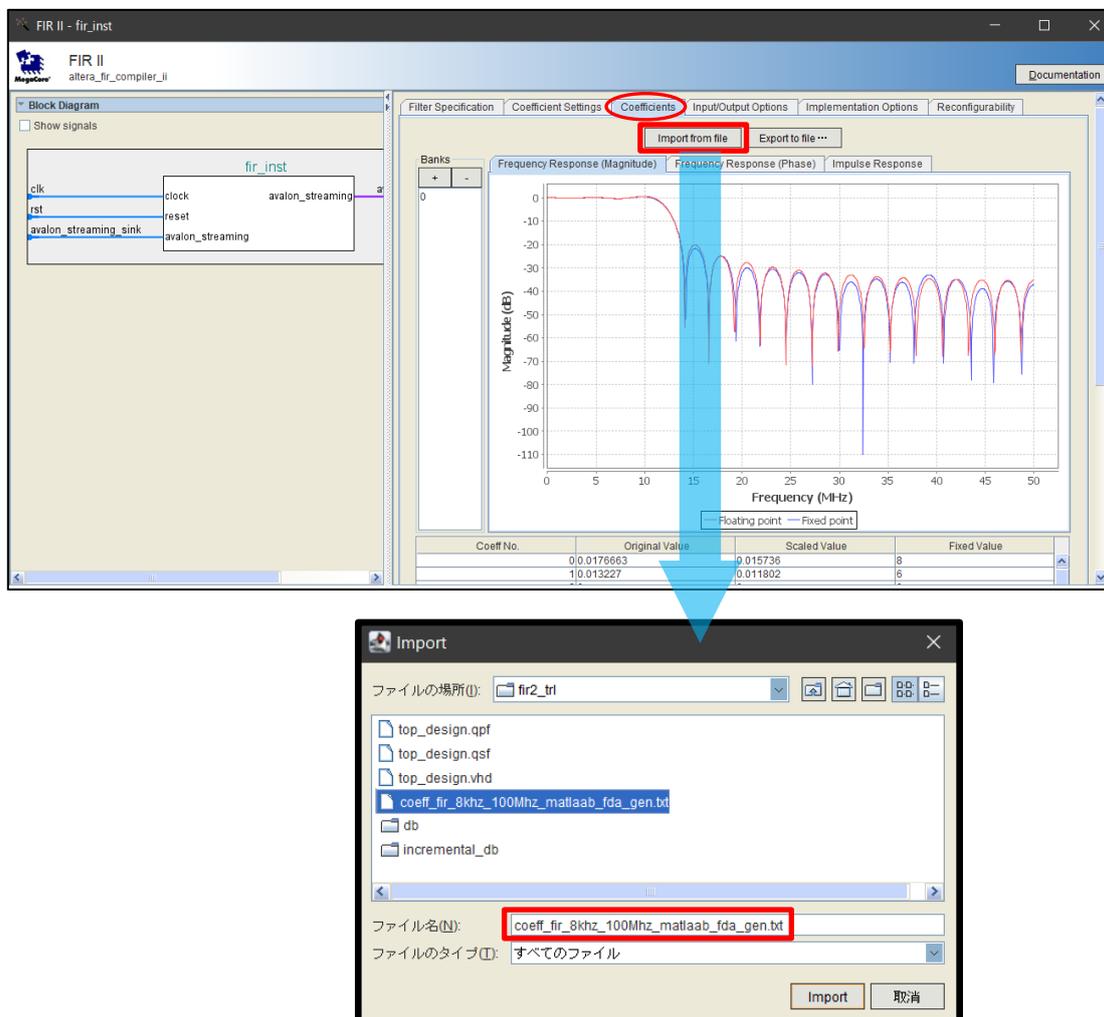
IP コアが生成する HDL 言語のファイル名(インスタンス名に該当)と HDL 言語の種類を指定して、**[OK]** ボタンをクリックします。

この資料では、インスタンス名を **fir_inst** とし、HDL 言語として VHDL を選択しますので、ファイル名には、**fir_inst.vhd** を指定します。



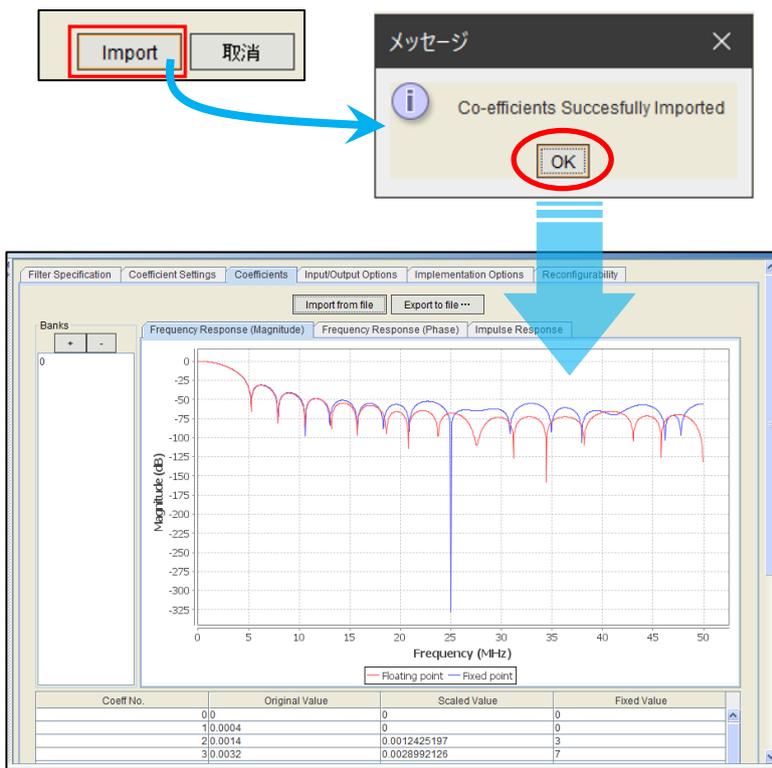
【図 4-3】 インスタンス名と IP コア生成言語の指定

FIR II のパラメータ設定画面が起動したら、**Coefficient** タブを選択後、**[Import file]** ボタンをクリックして、係数データを指定します。この資料では、係数データとして `coeff_fir_8khz_100Mhz_matlaab_fda_gen.txt` を選択します。



【図 4-4】係数データの取り込み ①

[Import] ボタンをクリックして、メッセージ画面が起動後、[OK] ボタンのクリックを行うことにより、図 4-5 のようにツールが係数データを取り込みます。



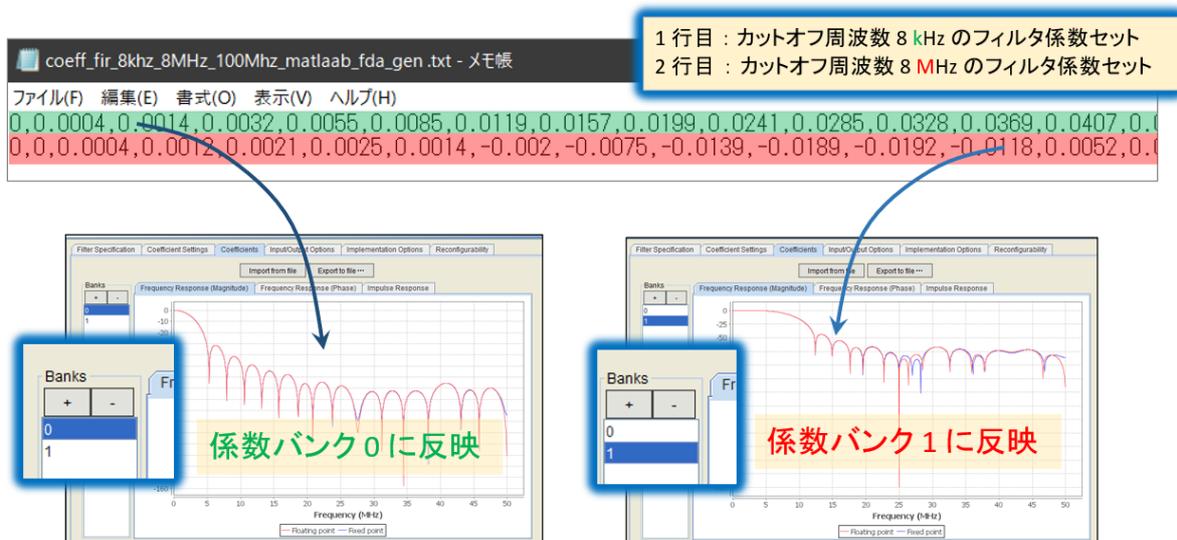
【図 4-5】係数データの取り込み ②

④ Point:

このとき、特性図がデフォルトから新たに取り込んだ内容に変化します。

⚠ 注記:

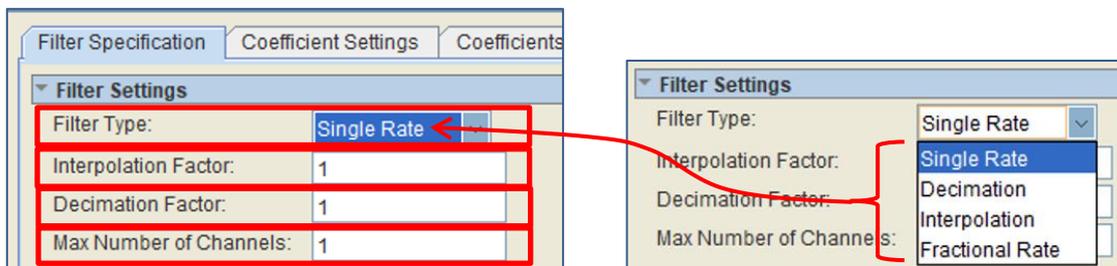
テキストファイルに、係数データ・セットを追加した場合、行数分の係数バンクとして取り込まれます。



【図 4-6】係数データの取り込み ③

5. フィルター形態の指定

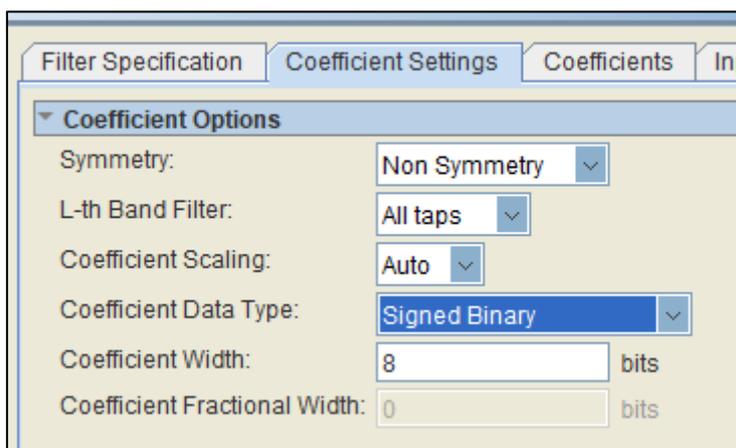
Filter Specification タブを選択します。**Filter Settings** 欄の **Filter Type** には、プルダウン・メニューから **Single Rate** を選択します。この資料の例では、補間フィルターや間引きフィルターとしては使用しないので、**Interpolation Factor** と **Decimation Factor** は、それぞれ **1** に設定します。チャンネル数 **Number of Channels** は、**1** に設定します。



【図 5-1】フィルター関連のパラメータ設定 ①

Coefficient Settings タブを選択します。**Coefficient Option** 欄の **Symmetry** は **Non Symmetry** を選択します。

L-th Band Filter は **All taps** を選択します。**Coefficient Scaling** は **Auto** を選択し、**Coefficient Data Type** は **Signed Binary** を選択します。



【図 5-2】フィルター関連のパラメータ設定 ②

FIR II は、係数データを整数に換算して FPGA に実装する仕様なので、取り込んだ係数データは、指定したビット幅で表現可能な範囲に応じた数値に細かくスケールした後、固定小数点(整数)に換算されます。

④ Point:

MATLAB のフィルター解析ツールが生成した係数データは、小数点以下の数値で表現するため、浮動小数点が適用されます。

④ Point:

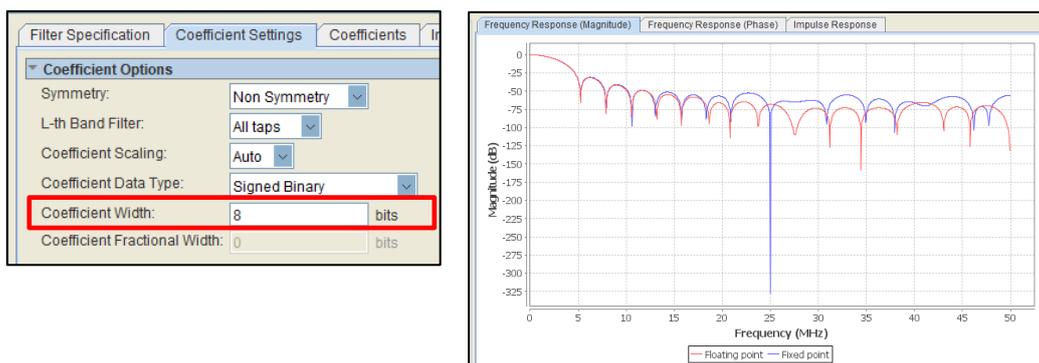
整数は、小数以下の桁が 0 の固定小数点です。

Coefficient タブを選択後、さらに **Frequency Response (Magnitude)** タブを選択します。デフォルト 8 ビットの固定小数点の係数データ向けにスケーリングした特性が、赤線の **Floating point** で表示されています

周波数特性の青線 **Fixed point** は、スケーリングされた浮動小数点の係数データを、固定小数点に換算したものです。

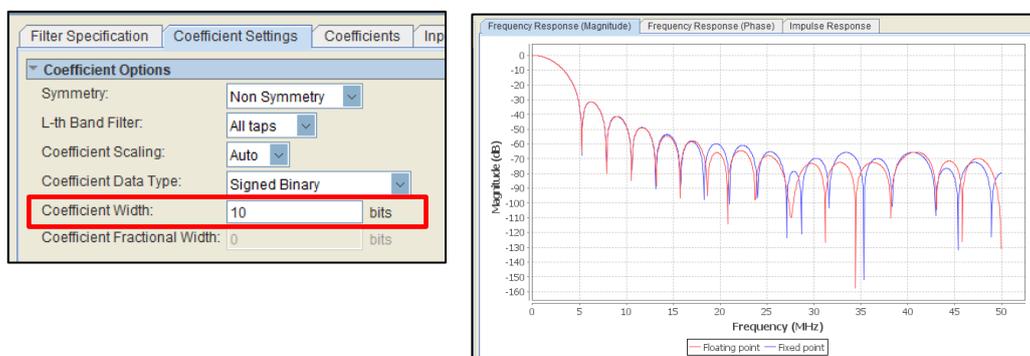
ビット幅を増やして固定小数点を浮動小数点に近づけるように、手作業で、青線を赤線に近づけるように調整します。

Coefficient Bit Width :8 に設定 ⇒ 係数が固定小数点 8 ビットの場合(デフォルト)



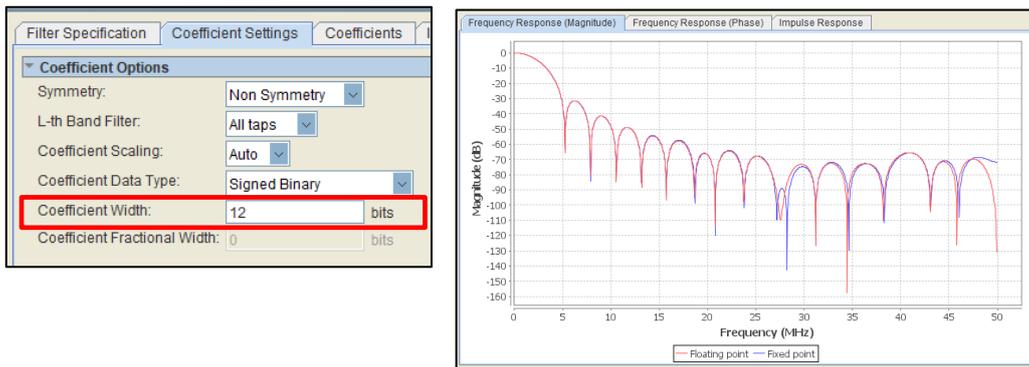
【図 5-3】係数データのビット幅の指定 ①

Coefficient Bit Width :10 に設定 ⇒ 係数が固定小数点 10 ビットの場合



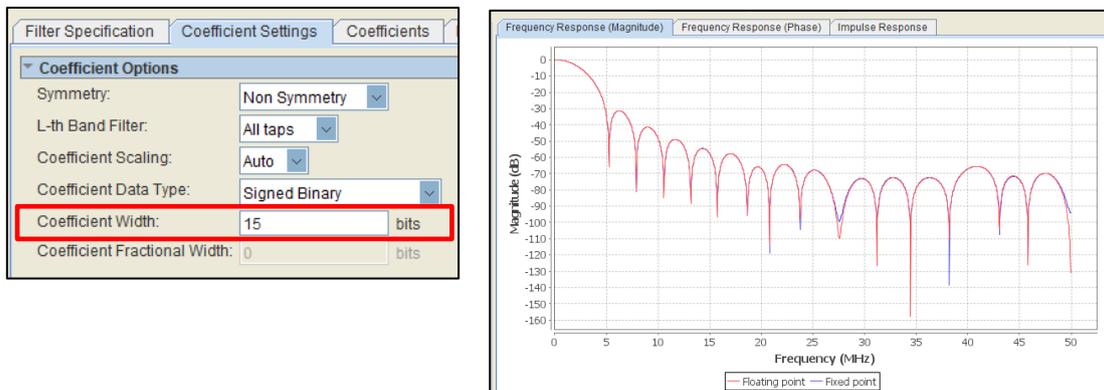
【図 5-4】係数データのビット幅の指定 ②

Coefficient Bit Width :12 に設定 ⇒ 係数の精度が固定小数点 12 ビットの場合



【図 5-5】係数データのビット幅の指定 ③

Coefficient Bit Width :15 に設定 ⇒ 係数の精度が固定小数点 15 ビットの場合



【図 5-6】係数データのビット幅の指定 ④

この資料では、係数が固定小数点 15 ビットで調整した場合の特性を採用します。

6. 入出力仕様の設定

Input/Output Options タブを選択後、この資料では *Input Options* 欄の *Input Data Type* に *Signed Binary* を選択し、*Input Bit Width* には **12** を指定します。

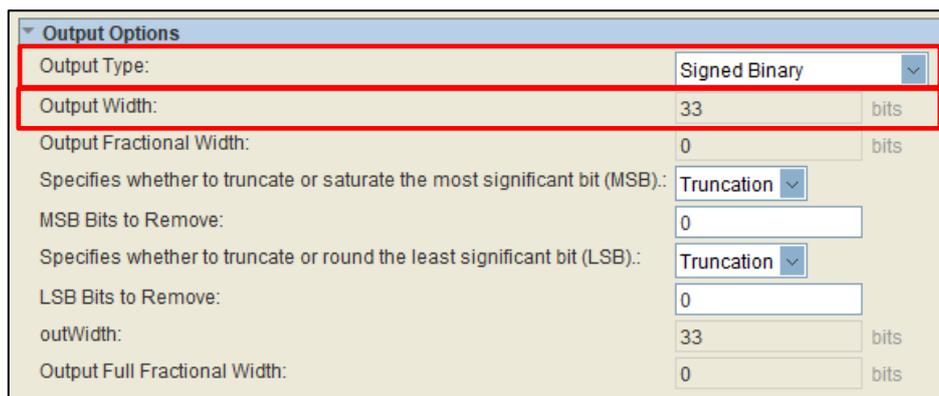


【図 6-1】データ入力信号に対する設定

この状態では *Output Options* 欄の *Output Full Bit Width* がグレーアウトされた状態で **33** を示していることが確認できます。この値は、前述の調整を手作業で実施した際、*Coefficient Bit Width* を **15** に指定したときにツールが自動算出したビット幅であり、出力データの分解能が最大となります。

Coefficient Bit Width の設定を変更すると、この値も連動して変更されます。

この資料では、*Output Data Type* を *Signed Binary* に指定します。



【図 6-2】データ出力信号に対する設定

7. 出力ビット幅の切り出しと端数処理

出力データに対して、必要に応じて、切り捨てや丸め処理などの端子処理や、ビットの切り出しを行い、冗長なデータ本数を削減することが出来ます。この資料では、33 ビットの出力データのうち、9 ビット目から 20 ビット目にある 12 ビットを、出力ポートに切り出す例を取り上げます。

その際、切り出す前の出力 33 ビット幅の下位 8 ビット側には丸め処理 (Round) を行います。また、上位 13 ビット側には飽和処理 (Saturating) を適用して、切り出す前の 33 ビット幅の出力が、切り出すビット幅（この場合、12 ビットが該当します。）で表現可能な最大値を超えた場合、切り出した出力ポートの値を強制的に、12 ビットで表現可能な最大値に調整します。

出力ビットの切り出し

33 ビットの内、上位 13 ビットと下位 8 ビットを削って、残りの 12 ビットを出力に反映させるには、**MSB Bits to Remove** を 13 に指定して、**LSB Bits to Remove** を 8 に指定します。この操作に連動して、グレーアウトされた状態でも、**Output Bit Width** が 12 に自動的に変更されます。

Output Options	
Output Type:	Signed Binary
Output Width:	12 bits
Output Fractional Width:	0 bits
Specifies whether to truncate or saturate the most significant bit (MSB):	Truncation
MSB Bits to Remove:	13
Specifies whether to truncate or round the least significant bit (LSB):	Truncation
LSB Bits to Remove:	8
outWidth:	33 bits
Output Full Fractional Width:	0 bits

【図 7-1】 出力データ幅の切り出し

端数処理

Specifies whether to truncate or saturate the most significant bit (MSB). を **Saturating** に指定して、上位ビット側の端数を飽和処理に設定します。

Specifies whether to truncate or round the least significant bit (LSB). を **Rounding** に指定して、下位ビット側の端数を丸め処理に設定します。

Output Options	
Output Type:	Signed Binary
Output Width:	12 bits
Output Fractional Width:	0 bits
Specifies whether to truncate or saturate the most significant bit (MSB):	Saturating
MSB Bits to Remove:	13
Specifies whether to truncate or round the least significant bit (LSB):	Rounding
LSB Bits to Remove:	8
outWidth:	33 bits
Output Full Fractional Width:	0 bits

【図 7-2】 出力データの端数処理

8. 回路規模および実装関連の設定

Filter Specification タブを選択します。**Filter Settings** 欄では、シングル・チャンネルのシングル・レート・フィルターとして設定しているので、**Frequency Specification** 欄の **Clock Rate** には **100 MHz**、**Input Sample Rate (MSPS)** には **100** を入力して、FIR フィルターのサンプリング周波数 $F_s = 100\text{MHz}$ の情報を与えます。

それ以外のパラメータについては、この資料では、デフォルトの値を適用します。

The image shows a software configuration window titled "Filter Specification" with three tabs: "Filter Specification", "Coefficient Settings", and "Coefficients". The "Filter Specification" tab is active. It contains several sections:

- Filter Settings:** Filter Type: Single Rate (dropdown), Interpolation Factor: 1, Decimation Factor: 1, Max Number of Channels: 1.
- Frequency Specification:** Clock Rate: 100 MHz (input field), Clock Slack: 0 MHz (input field), Input Sample Rate (MSPS): 100 (input field). The "Clock Rate" and "Input Sample Rate" fields are highlighted with red boxes.
- Coefficient Reload Options:** Coefficients Reload (checkbox, unchecked), Base Address: 0 (input field), Read/Write Mode: Read/Write (dropdown).
- Flow Control:** Back Pressure Support (checkbox, unchecked).

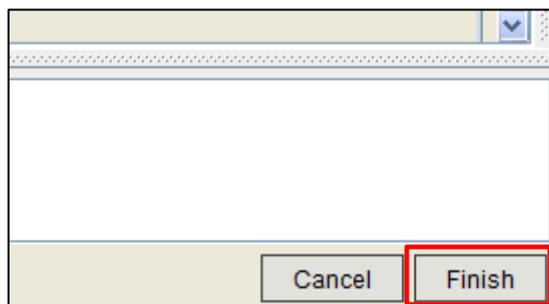
【図 8-1】動作周波数とサンプリング周波数の設定

Implementation Option タブを選択して、実装するデバイス・ファミリーなど、リソースに関連した情報が確認できます。この資料では、デフォルトの値を適用します。

Category	Parameter	Value
Resource Optimization Settings	Device Family:	Cyclone 10 LP
	Speed Grade:	Medium
	Memory Block Threshold:	20
	Dual Port RAM Threshold:	1280
	Large RAM Threshold:	1000000
	Hard Multiplier Threshold:	-1
Resource Estimation	Number of LUTs:	2187
	Number of DSPs:	26
	Number of Memory Bits:	0

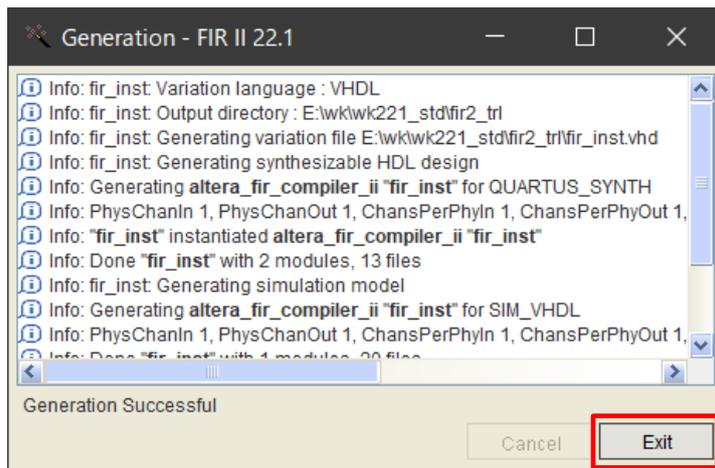
【図 8-2】実装関連の設定 ①

右下の **[Finish]** ボタンをクリックして、IP コアの HDL の生成を行います。



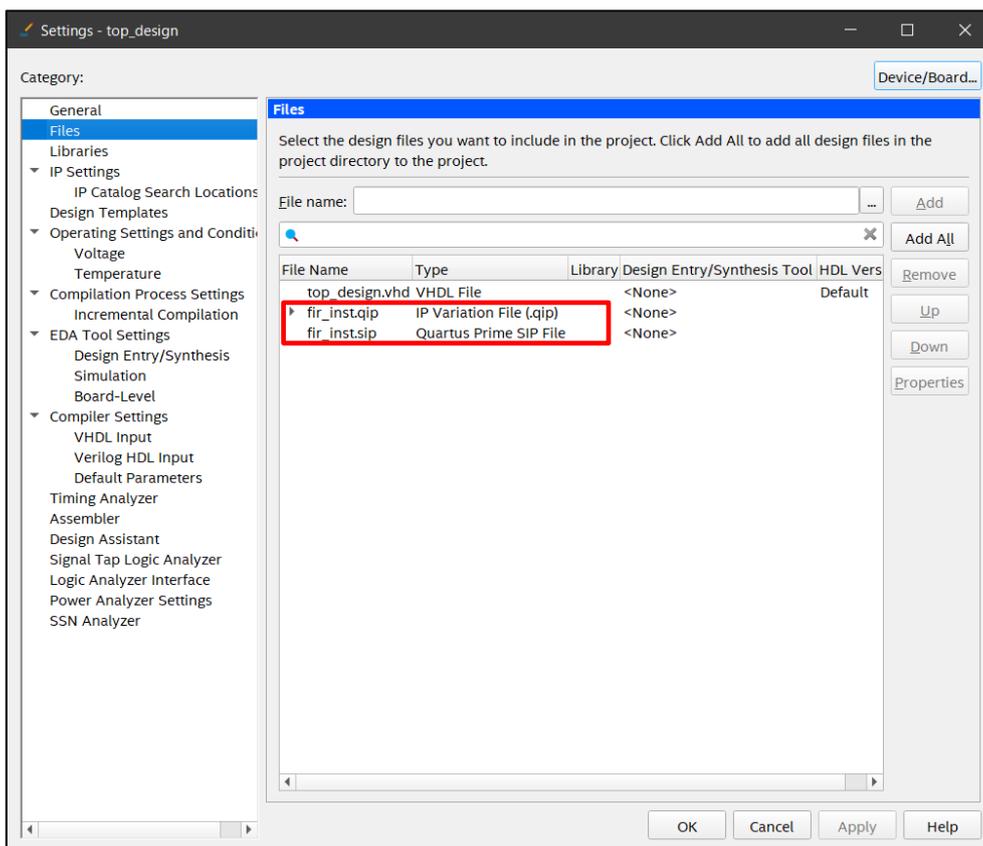
【図 8-3】実装関連の設定 ②

HDL の生成が終了したら、[Exit] ボタンをクリックします。



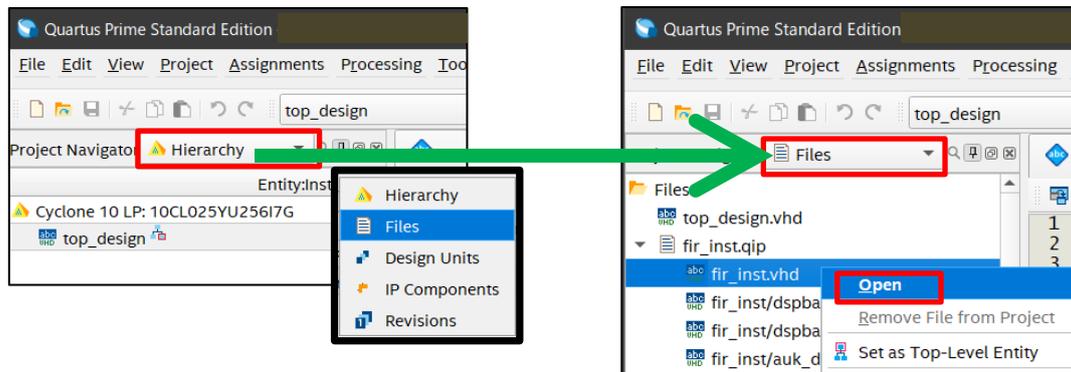
【図 8-4】実装関連の設定 ③

Assignments メニューから **Settings** を選択後、**Settings** 画面内の **Category** 欄で **Files** をハイライトして、右側の **Files** 欄に、**fir_inst.qip** ファイルと **fir_inst.sip** ファイルが追加されていることを確認します。



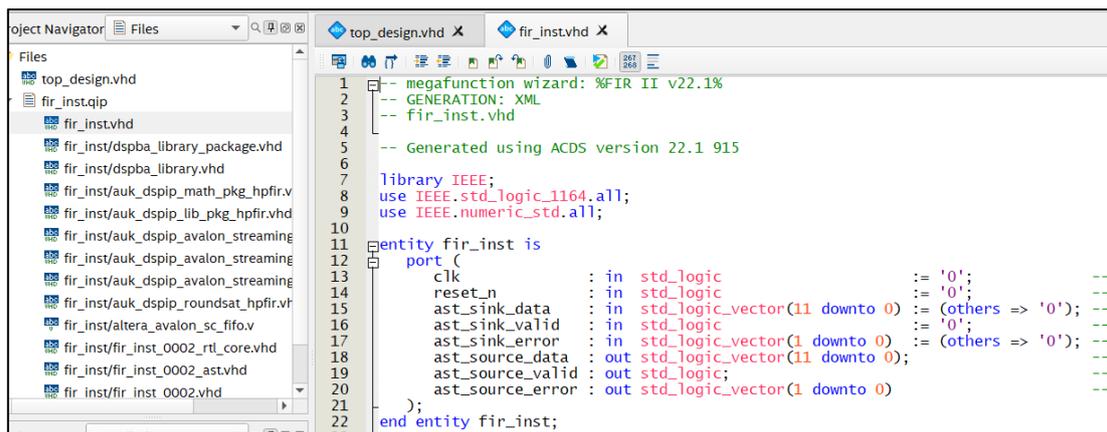
【図 8-5】実装関連の設定 ④

Quatus® Prime の Project Navigator の右にあるプルダウン・メニューで *Hierarchy* を *Files* に変更した後、*Files* 欄の *fir_inst.qip* の下に表示される *fir_inst.vhd* をハイライトして、右クリックから *Open* を選択します。



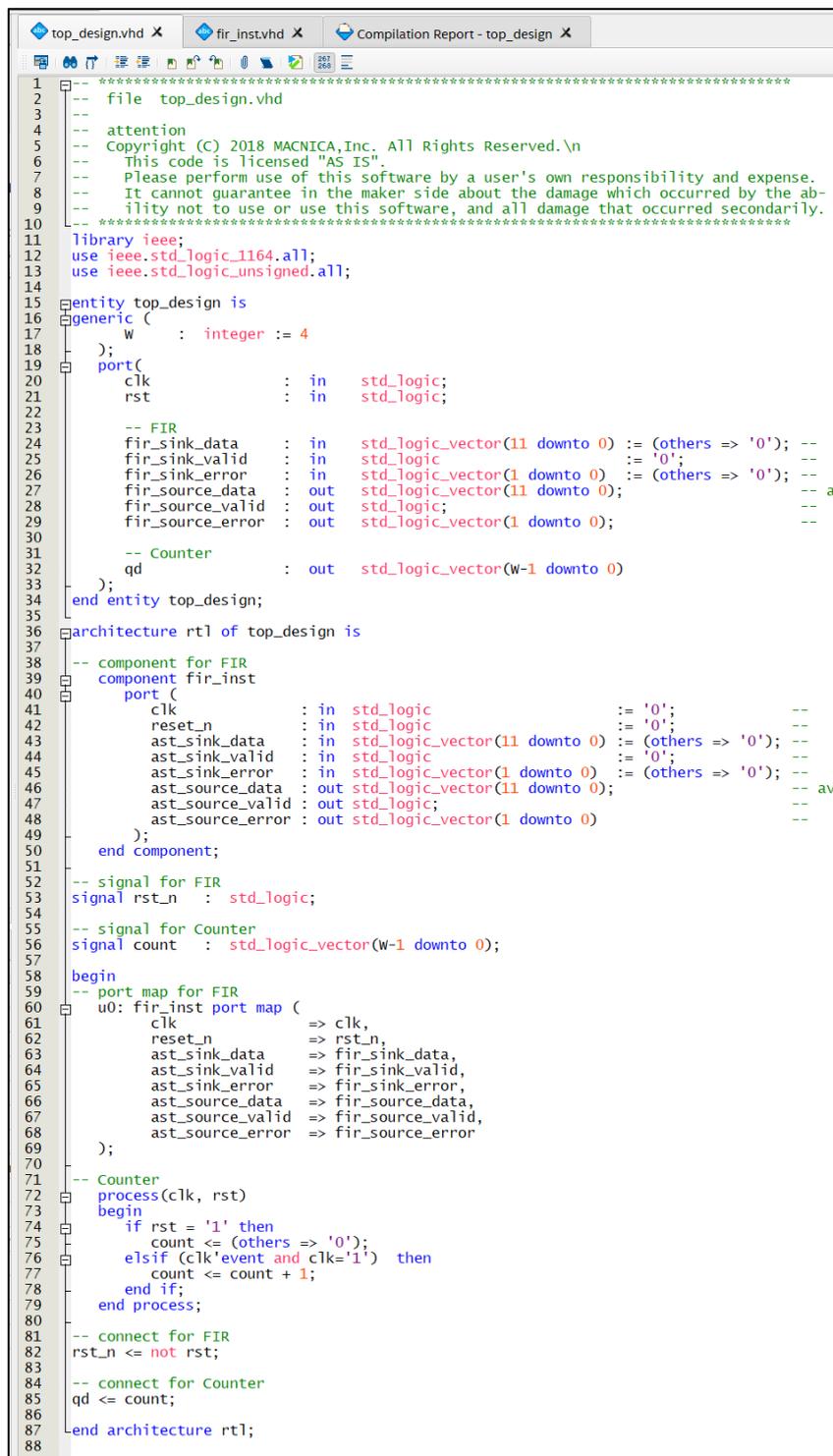
【図 8-6】実装関連の設定 ⑤

fir_inst.vhd が表示されます。



【図 8-7】実装関連の設定 ⑥

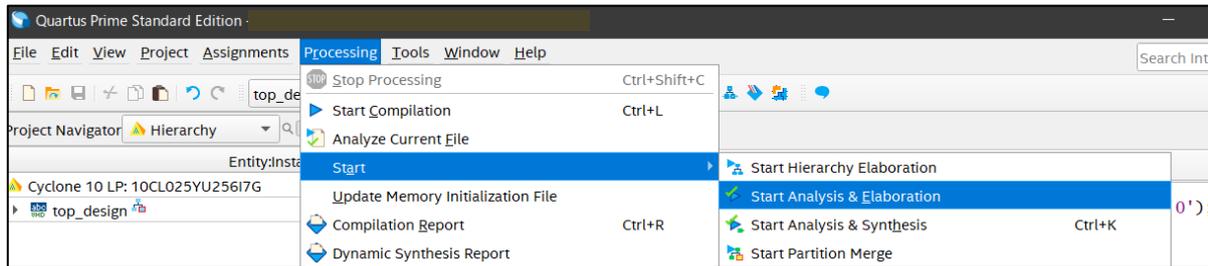
手作業で、下位階層のインスタンス *fir_inst* を最上位階層に適切に接続します。



```
1  -- file top_design.vhd
2  --
3  --
4  -- attention
5  -- Copyright (C) 2018 MACNICA, Inc. All Rights Reserved.\n
6  -- This code is licensed "AS IS".
7  -- Please perform use of this software by a user's own responsibility and expense.
8  -- It cannot guarantee in the maker side about the damage which occurred by the ab-
9  -- ility not to use or use this software, and all damage that occurred secondarily.
10 -- *****
11 library ieee;
12 use ieee.std_logic_1164.all;
13 use ieee.std_logic_unsigned.all;
14
15 entity top_design is
16 generic (
17     W      : integer := 4
18 );
19 port(
20     clk      : in    std_logic;
21     rst      : in    std_logic;
22
23     -- FIR
24     fir_sink_data  : in    std_logic_vector(11 downto 0) := (others => '0'); --
25     fir_sink_valid : in    std_logic                    := '0'; --
26     fir_sink_error : in    std_logic_vector(1 downto 0) := (others => '0'); --
27     fir_source_data : out   std_logic_vector(11 downto 0); -- a
28     fir_source_valid : out   std_logic; --
29     fir_source_error : out   std_logic_vector(1 downto 0); --
30
31     -- Counter
32     qd      : out   std_logic_vector(W-1 downto 0)
33 );
34 end entity top_design;
35
36 architecture rtl of top_design is
37
38     -- component for FIR
39     component fir_inst
40     port (
41         clk      : in    std_logic                    := '0'; --
42         reset_n  : in    std_logic                    := '0'; --
43         ast_sink_data  : in    std_logic_vector(11 downto 0) := (others => '0'); --
44         ast_sink_valid : in    std_logic                    := '0'; --
45         ast_sink_error : in    std_logic_vector(1 downto 0) := (others => '0'); -- av
46         ast_source_data : out   std_logic_vector(11 downto 0); --
47         ast_source_valid : out   std_logic; --
48         ast_source_error : out   std_logic_vector(1 downto 0) --
49     );
50     end component;
51
52     -- signal for FIR
53     signal rst_n : std_logic;
54
55     -- signal for Counter
56     signal count : std_logic_vector(W-1 downto 0);
57
58     begin
59     -- port map for FIR
60     u0: fir_inst port map (
61         clk      => clk,
62         reset_n  => rst_n,
63         ast_sink_data  => fir_sink_data,
64         ast_sink_valid => fir_sink_valid,
65         ast_sink_error => fir_sink_error,
66         ast_source_data => fir_source_data,
67         ast_source_valid => fir_source_valid,
68         ast_source_error => fir_source_error
69     );
70
71     -- Counter
72     process(clk, rst)
73     begin
74         if rst = '1' then
75             count <= (others => '0');
76         elsif (clk'event and clk='1') then
77             count <= count + 1;
78         end if;
79     end process;
80
81     -- connect for FIR
82     rst_n <= not rst;
83
84     -- connect for Counter
85     qd <= count;
86
87 end architecture rtl;
88
```

【図 8-8】実装関連の設定 ⑦

Processing メニューから、**Start** を選択後、**Start Analysis & Elaboration** を選択します。下位階層との接続が適切な場合、エラーに遭遇することなく、**Start Analysis & Elaboration** が終了します。



【図 8-9】実装関連の設定 ⑧

Start Analysis & Elaboration 終了後、ピン・アサインなどの諸設定を行い、SDC ファイルなどの追加が終了したら、**Processing** メニューから、**Start Compilation** を選択して、フル・コンパイルを行います。

⚠ 注記:

この簡易チュートリアルでは、FPGA の設計に不可欠な RTL シミュレーション、ゲートレベル・シミュレーション、ピン・アサイン、タイミング解析などのフローは、便宜上、割愛しています。

9. まとめ

この資料を利用することで、係数生成の過程を含めて、FIR II IP を使用したデザインの一連の操作を簡易的に実施することができます。

改版履歴

Revision	年月	概要
1	2016年2月	初版
2	2023年12月	・アルテラ®社に関連した用語をインテル®社向けに刷新 ・MATLAB Online に関連した内容を追加

免責およびご利用上の注意

弊社より資料を入手されましたお客様におかれましては、下記の使用上の注意を一読いただいた上でご使用ください。

1. 本資料は非売品です。許可無く転売することや無断複製することを禁じます。
2. 本資料は予告なく変更することがあります。
3. 本資料の作成には万全を期していますが、万一ご不明な点や誤り、記載漏れなどお気づきの点がありましたら、本資料を入手されました下記代理店までご一報いただければ幸いです。
[株式会社マクニカ 半導体事業 お問い合わせフォーム](#)
4. 本資料で取り扱っている回路、技術、プログラムに関して運用した結果の影響については、責任を負いかねますのであらかじめご了承ください。
5. 本資料は製品を利用する際の補助的な資料です。製品をご使用になる際は、各メーカー発行の英語版の資料もあわせてご利用ください。