DG0514
Demo Guide
IGLOO2 FPGA Adaptive FIR Filter - Libero SoC v11.8

Microsemi

Power Matters.™

Microsemi makes no warranty, representation, or guarantee regarding the information contained herein or the suitability of its products and services for any particular purpose, nor does Microsemi assume any liability whatsoever arising out of the application or use of any product or circuit. The products sold hereunder and any other products sold by Microsemi have been subject to limited testing and should not be used in conjunction with mission-critical equipment or applications. Any performance specifications are believed to be reliable but are not verified, and Buyer must conduct and complete all performance and other testing of the products, alone and together with, or installed in, any end-products. Buyer shall not rely on any data and performance specifications or parameters provided by Microsemi. It is the Buyer's responsibility to independently determine suitability of any products and to test and verify the same. The information provided by Microsemi hereunder is provided "as is, where is" and with all faults, and the entire risk associated with such information is entirely with the Buyer. Microsemi does not grant, explicitly or implicitly, to any party any patent rights, licenses, or any other IP rights, whether with regard to such information itself or anything described by such information. Information provided in this document is proprietary to Microsemi, and Microsemi reserves the right to make any changes to the information in this document or to any products and services at any time without notice.

About Microsemi

Microsemi Corporation (Nasdaq: MSCC) offers a comprehensive portfolio of semiconductor and system solutions for aerospace & defense, communications, data center and industrial markets. Products include high-performance and radiation-hardened analog mixed-signal integrated circuits, FPGAs, SoCs and ASICs; power management products; timing and synchronization devices and precise time solutions, setting the world's standard for time; voice processing devices; RF solutions; discrete components; enterprise storage and communication solutions, security technologies and scalable anti-tamper products; Ethernet solutions; Power-over-Ethernet ICs and midspans; as well as custom design capabilities and services. Microsemi is headquartered in Aliso Viejo, California, and has approximately 4,800 employees globally. Learn more at www.microsemi.com.

50200514. 7.0 3/17

Contents

Figures

Tables

1 Revision History

The revision history describes the changes that were implemented in the document. The changes are listed by revision, starting with the most current publication.

1.1 Revision 7.0

Updated the document for Libero v11.8 software release.

1.2 Revision 6.0

Updated the document for Libero v11.7 software release.

1.3 Revision 5.0

Updated the document for Libero v11.6 software release.

1.4 Revision 4.0

Updated the document for Libero v11.5 software release.

1.5 Revision 3.0

Updated the document for Libero v11.4 software release.

1.6 Revision 2.0

The following is a summary of the changes in revision 2.0 of this document.

- Updated the document for Libero v11.3 software release.
- The Theory of Operation, page 2 was updated.
- Updated Figure 7, page 8.

1.7 Revision 1.0

Revision 1.0 was the first publication of this document.

2 IGLOO2 FPGA Adaptive FIR Filter Demo

2.1 Introduction

The IGLOO2 FPGA devices integrate a fourth generation flash-based FPGA fabric architecture, which includes embedded math blocks optimized specifically for digital signal processing (DSP) applications such as, finite impulse response (FIR) filters, infinite impulse response (IIR) filters, and fast Fourier transform (FFT) functions.

An adaptive filter is a filter that automatically adjusts the filter coefficients according to the underlying adaptive algorithm and the input signal characteristics. Due to its self adjustment of transfer function of an unknown system and computational requirements, adaptive filters are widely used in different areas of DSP application such as communication, biomedical instrumentation, audio processing, and video processing.

The least mean square (LMS) is the basic adaptive algorithm used in adaptive filters to update the filter coefficients. The LMS algorithm has advantages over other algorithms because of its simplicity, less computations and best performance in terms of the number of iterations required for convergence.

In this demo, the suppression of a narrow band signal interference on a wide band signal is implemented using an IGLOO2 device. See the following figure.

The LMS algorithm is implemented in the FPGA fabric to adjust the filter coefficients based on the mean square error (MSE) approach. CoreFIR IP is used to perform the filtering operation and CoreFFT IP is used to generate the output spectrum to observe that the narrow band interfering signal component is suppressed. The host interface is implemented in FPGA fabric using CoreUART IP to communicate with the host PC. A user friendly IGL2_Adaptive_FIR_Filter.exe generates input signals (narrow band signal and wide band signal), and also plots the input or output waveforms and the required spectrum.

Figure 1 • **Narrowband Interference Cancellation**

Input Signal (Wideband + Narrowband)   IGLOO2 Evaluation Kit   Output Signal (Wideband)
(Adaptive FIR Filter)
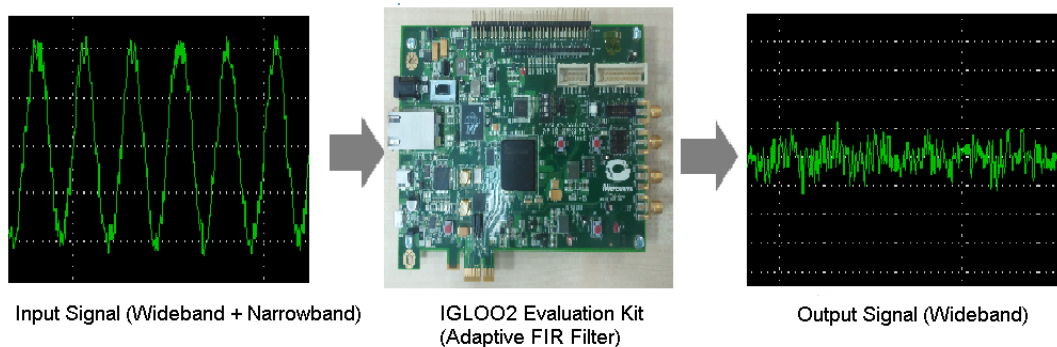
2.1.1 Theory of Operation

Adaptive filters are mainly categorized into four basic architectures:

- System identification
- Noise cancellation
- Linear prediction
- Inverse modeling

In this demo, linear prediction architecture is used to implement adaptive filter. The LMS algorithm uses a gradient search technique to determine the filter coefficients that minimize the mean square prediction error. The estimate of the gradient is based on the sample values of the tap-input vector and the error signal. The algorithm iterates over each coefficient in the filter, moving it in the direction of the approximated gradient. After reaching the optimal filter coefficients, the error signal $e(n)$ consists of the Wide band signal. The following figure shows the linear prediction based adaptive filter architecture.

Figure 2 • **Linear Prediction Adaptive Filter Architecture**
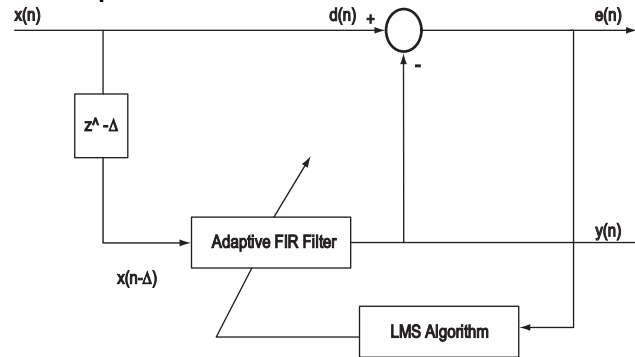
The input signal x(n) consists of wide band signals along with the narrow band signals that are not required, see Figure 3, page 4. In a linear prediction architecture, the desired signal d(n) is same as the input signal x(n) and a delayed input x(n-D) is fed to the adaptive filter as shown in the preceding figure. The delay factor D (delta) de-correlates the wide band component and correlates the narrow band component of the desired signal d(n) with the delayed input signal x(n-D).

The adaptive filter tries to estimate the narrow band component y(n), and forms an equivalent transfer function, which is similar to that of narrow band filters centered at the frequencies of the narrow band components of the input signal. At the summing junction, the filtered input signal subtracting with delayed input signal produces an error signal. The error signal is used by the LMS algorithm to adjust the filter coefficients. After some iterations, the Error signal converges to a wide band component.

The following equations describe computing the coefficients using the LMS algorithm.

$$y(n) = \sum_{k=0}^{k=1-1} h(n) \times x(n - \Delta - k)$$

where,

According to EQ 1, narrow band component y(n), is the adaptive filter output

h(n) is the filter coefficients

x(n-Δ) is the input signal to the adaptive filter

l is the length of the filter (number of taps)

k is the index variable

The error is computed using the following equation:

e(n) = d(n) - y(n)

where,

e(n) is the error signal

d(n) is desired signal

The filter coefficients are updated using the following equation:

h(n+1) = h(n)+μ*e(n)*x(n-Δ)

where,

h(n+1) is the estimated filter coefficients

h(n) is present filter coefficients

μ is the step size factor

Figure 3 • **Input Spectrum of Narrow Band Signal + Wide Band Signal**
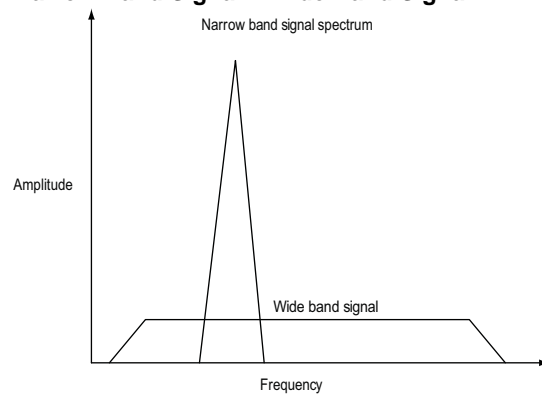
Figure 4 • **Output Spectrum of Wide Band Signal**
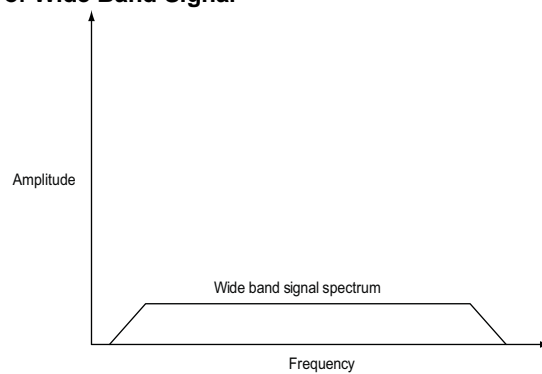
2.2   Design Requirements

Table 1 • **Design Requirements**

Design Requirements	Description
Hardware Requirements	
IGLOO2 Evaluation Kit: – FlashPro4 programmer – USB A to Mini-B cable	Rev C or later
Host PC or Laptop	Windows 7 64-bit Operating System
Software Requirements	
Libero® System-on-Chip (SoC)	v11.8
FlashPro Programming Software	v11.8
Host PC Drivers	USB to UART drivers
Framework	Microsoft NET Framework 4 Client for launching demo GUI

2.3 Demo Design

2.3.1 Introduction

The design files for this demo can be downloaded from the Microsemi website:
http://soc.microsemi.com/download/rsc/?f=m2gl_dg0514_adaptive_fir_filter_liberov11p8_df

Design files include:

- Design files
- GUI
- Programming files
- Readme file

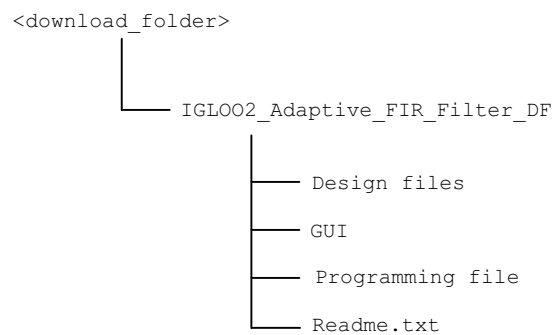The following figure shows the top-level structure of the design files. For further details, see the Readme.txt file.

Figure 5 • **Demo Design Files Top-Level Structure**

```
<download_folder>

        IGLOO2_Adaptive_FIR_Filter_DF

                Design files

                GUI

                Programming file

                Readme.txt
```

2.3.2 Demo Design Description

This demo design uses the following blocks:

- Data Handle Block (SmartDesign)
- Filter Control (user RTL)
- LMS_FIR_TOP (Smart Design)
- TPSRAM IP (IPcore)
- CoreFFT (IPcore)
- SYSRESET (IPcore)
- OSC (IPcore)
- CCC (IPcore)
- CoreUART (IP Core)

Figure 6 • **Adaptive FIR Filter Demo Design Block Diagram**
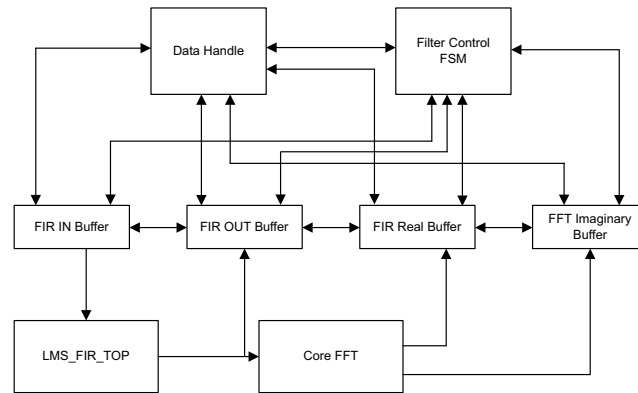
2.3.2.1 Data Handle Block

The Data Handle block consists of a CoreUART IP and UART interface finite state machine to handle the controls and operations between the host PC (GUI interface) and fabric logic. It controls loading filter input data to the corresponding input data buffer and then send and receive data from the host PC.

2.3.2.2 Filter Control

Controls the FIR filter and FFT operations. It loads the filtered data to the corresponding output buffer and moves the FFT output data to the corresponding output data buffer.

2.3.2.3 LMS_FIR_TOP

This is a SmartDesign block implemented in the fabric. It consists of the following blocks:

- **LMS_CONTROL_FSM**: This finite-state machine (FSM) is implemented in the RTL to provide the control signals to the LMS_ALGO block.
- **LMS_ALGO**: This LMS algorithm is implemented in RTL to compute the error signal, correction factor, filter coefficients, and to send the filter coefficients to the Core FIR filter.
- **CoreFIR**: CoreFIR IP is used in the Reloadable Coefficient mode to configure its coefficients on the fly. CoreFIR IP configuration is as follows:
  - Version: 8.6.101
  - Filter Type: Single rate fully enumerated
  - No of taps: 8
  - Coefficients type: Reloadable
  - Coefficients bit width: 16 (signed)
  - Data bit width: 16 (signed)
  - Filter structure: Transposed with no symmetry

2.3.2.4 TPSRAM IP

TPSRAM IP uses the following configurations:

- Input signal data buffer
- Output signal buffer
- Output signal FFT real data buffer
- Output signal FFT imaginary data buffer

Table 2 • **TPSRAM Configuration for Data Buffers**

Buffer	Write Port		Read Port	
	Depth	Width	Depth	Width
FIR Input Signal	2048	8	1024	16
FIR Output Signal	1024	16	1024	16

Table 2 •   **TPSRAM Configuration for Data Buffers**

FFT Output Real Signal	1024	16	1024	16
FFT Output Imaginary Signal	1024	16	1024	16

2.3.2.5   CoreFFT

CoreFFT IP is used to generate the frequency spectrum of the filtered data. CoreFFT IP configuration is as follows:

- Version: 6.4.105
- FFT Architecture: In place
- FFT type: Forward
- FFT Scaling: Conditional
- FFT Transform Size: 256
- Width: 16

2.3.2.6   SYSRESET

SYSRESET IP provides the power-on reset signal.

2.3.2.7   OSC

OSC IP is configured as an RC oscillator to provide the 50 MHz signal to the CCC (clock conditioning circuit), narrow band component y(n).

2.3.2.8   CCC

CCC IP is configured to provide a 100 MHz clock signal

For detailed SmartDesign implementation and resource usage summary, see Appendix: SmartDesign Implementation, page 24.

2.3.2.9   CoreUART

The CoreUART IP is used to transfer the data between the host PC (GUI) and the IGLOO2 device. The CoreUART Configuration is as follows:

- Version: 5.5.101
- TxFIFO: Disable
- RxFIFO: Disable
- RxLegacyMode: Disable
- Baud rate: 115200
- Number of bits: 8
- Stop bits: 1
- Parity: None

2.4   Setting Up the Demo Design

The following steps describe how to setup the hardware demo:

1. Connect the jumpers on the IGLOO2 Evaluation Kit board as shown in the following table.

*Table 3 •*   **IGLOO2 FPGA Evaluation Kit Jumper Settings**

Jumper	Pin (From)	Pin (To)	Comments
J22	1	2	Default
J23	1	2	Default
J24	1	2	Default
J8	1	2	Default
J3	1	2	Default

CAUTION: While making the jumper connections, the power supply switch **SW7** must be switched OFF.

2. Connect the Power supply to the **J6** connector, switch On the power supply switch, **SW7**.
3. Connect the FlashPro4 programmer to the **J5** connector of the IGLOO2 Evaluation Kit board.
4. Connect the host PC USB port to the **J18** USB connector on the IGLOO2 Evaluation Kit board using the USB mini-B cable.

The following figure shows the board setup for running the Adaptive FIR filter demo on the IGLOO2 Evaluation Kit.

Figure 7 • **IGLOO2 Evaluation Kit Adaptive FIR Filter Demo Setup**
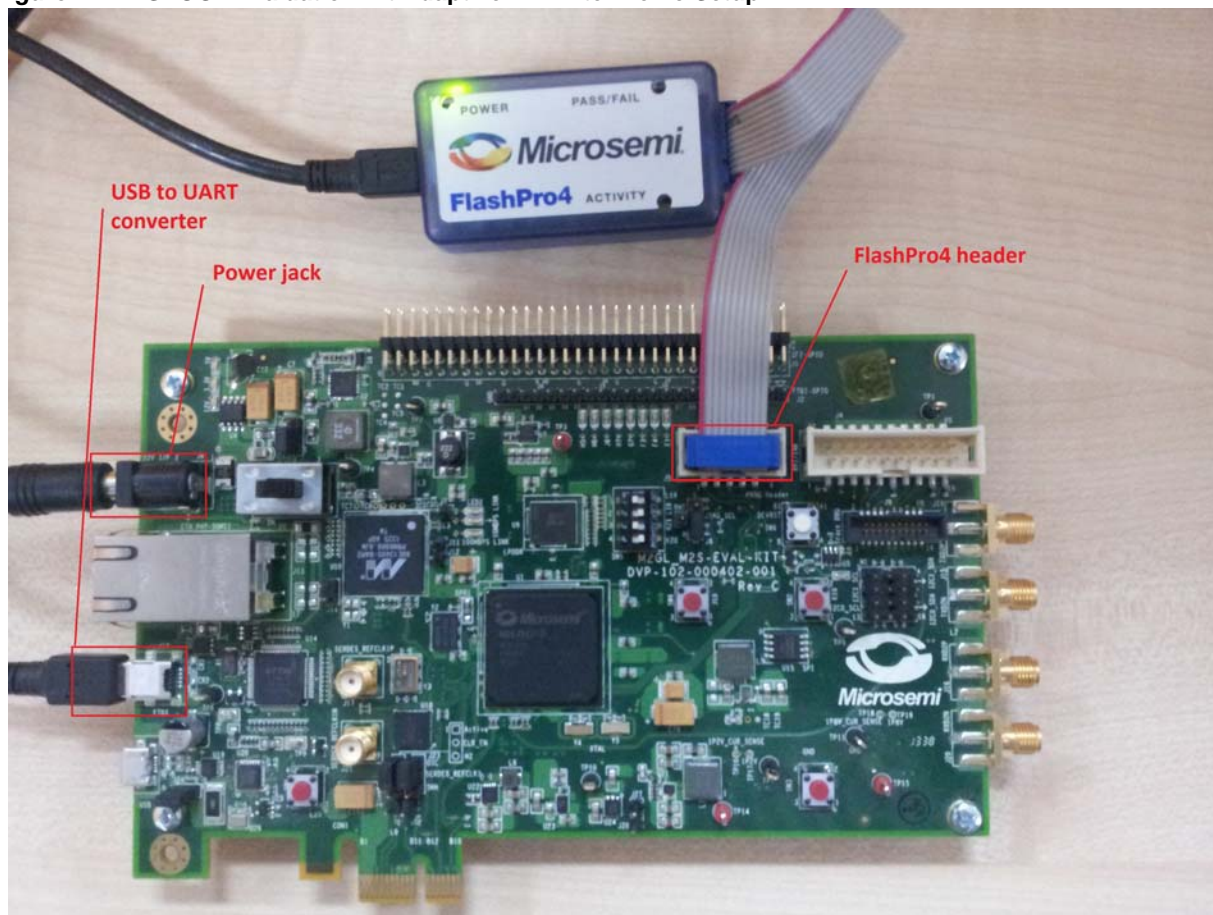
5. Ensure that the USB to UART bridge drivers are automatically detected. This can be verified in the Device Manager of the host PC. The FTDI USB to UART converter enumerates four COM ports. For USB 2.0, note down the USB Serial Converter D COM port number to use it in the `IGL2_Adaptive_FIR_Filter.exe`.

The following figure shows the USB 2.0 Serial port properties and the connected COM10 and USB Serial Converter D.

Figure 8 • **USB to UART Bridge Drivers**
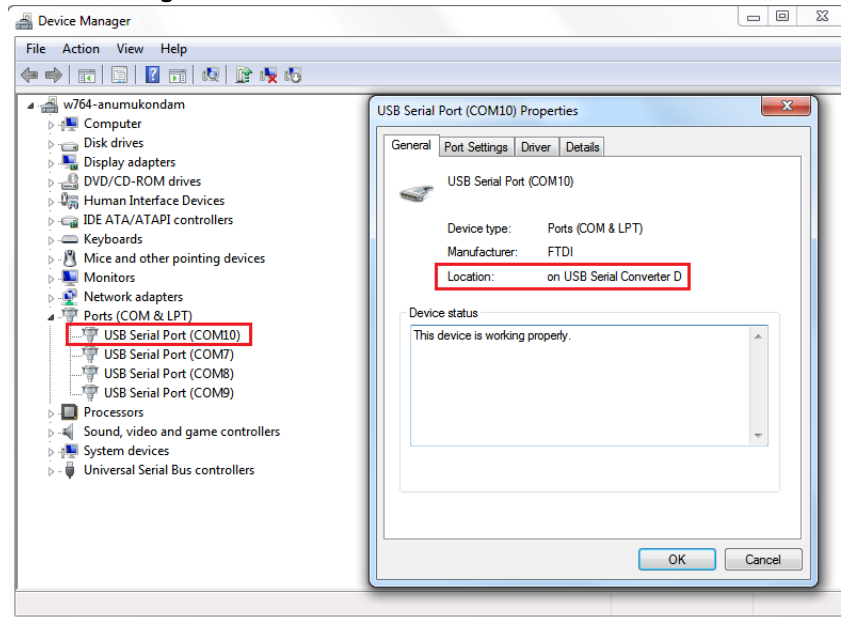
6. If the USB to UART bridge drivers are not installed, download and install the drivers from *www.microsemi.com/soc/documents/CDM_2.08.24_WHQL_Certified.zip*.

2.5 Programming the Demo Design

The following steps describe how to program the demo design:

1. Download the demo design from:
   http://soc.microsemi.com/download/rsc/?f=m2gl_dg0514_adaptive_fir_filter_liberov11p8_df
2. Launch the FlashPro software.
3. Click **New Project**.
4. In the **New Project** window, enter the project name as `IGL2_Adaptive_FIR_Filter`.

Figure 9 • **FlashPro - New Project**
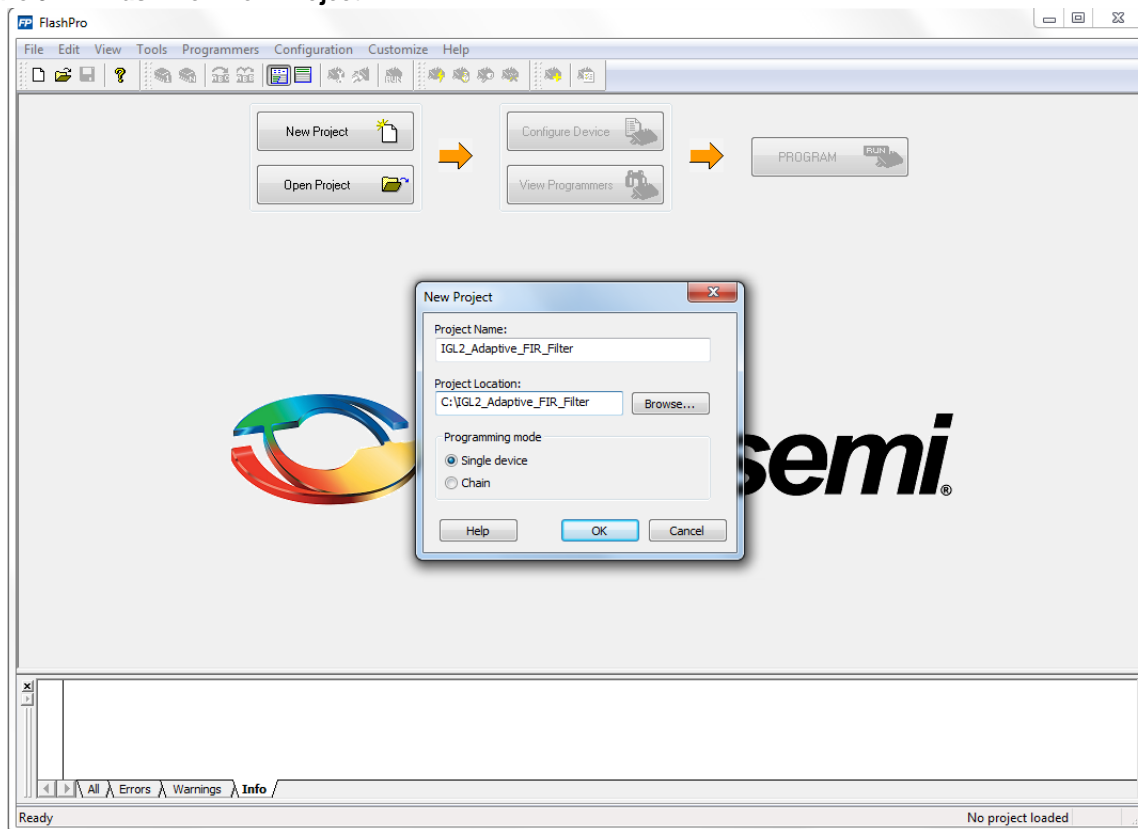
5. Click **Browse** and navigate to the location where you want to save the project.
6. Select **Single device** as the **Programming mode**.
7. Click **OK** to save the project.

2.5.1 Setting Up the Device

The following steps describe how to configure the device:

1. Click **Configure Device** on the FlashPro GUI.
2. Click **Browse** and navigate to the location where the `IGL2_Adaptive_FIR_Filter.stp` file is located and select the file. The default location of the programming file is:
   <download_folder>/IGLOO2_Adaptive_FIR_Filter_DF\Programmingfile\IGL2_Adaptive_FIR_Filter.stp.
3. Click **Open**. The required programming file is selected and is ready to be programmed in the device.
4. Select **Advanced** as Mode and **PROGRAM** as Action.

2.5.2 Programming the Device

*Figure 10 •* **FlashPro Project Configuration**
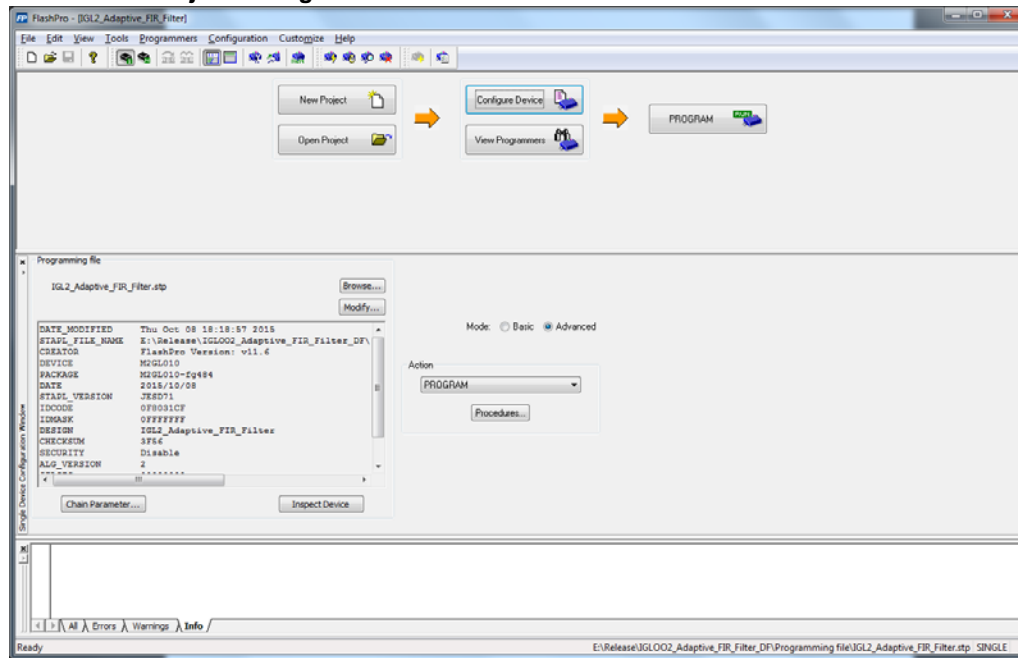
Click **PROGRAM** to start programming the device. Wait until Programmer Status is changed to **RUN PASSED**.

Figure 11 • **FlashPro Project RUN Passed**
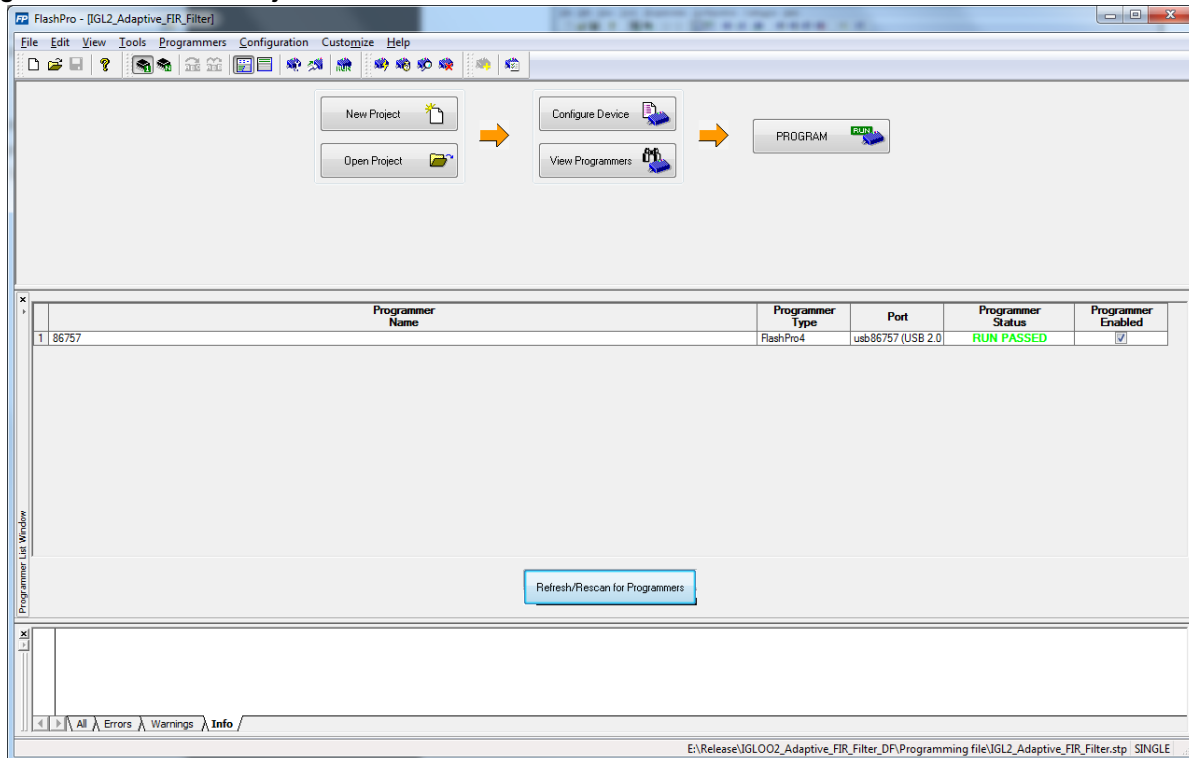
2.5.2.1 Adaptive FIR Filter Demo GUI

The adaptive FIR filter demo is provided with a user friendly GUI that runs on the host PC and communicates with IGLOO2 Evaluation Kit. UART is used as the communication protocol between the host PC and the IGLOO2 Evaluation Kit.

Figure 12 • **Adaptive FIR Filter Demo Window**
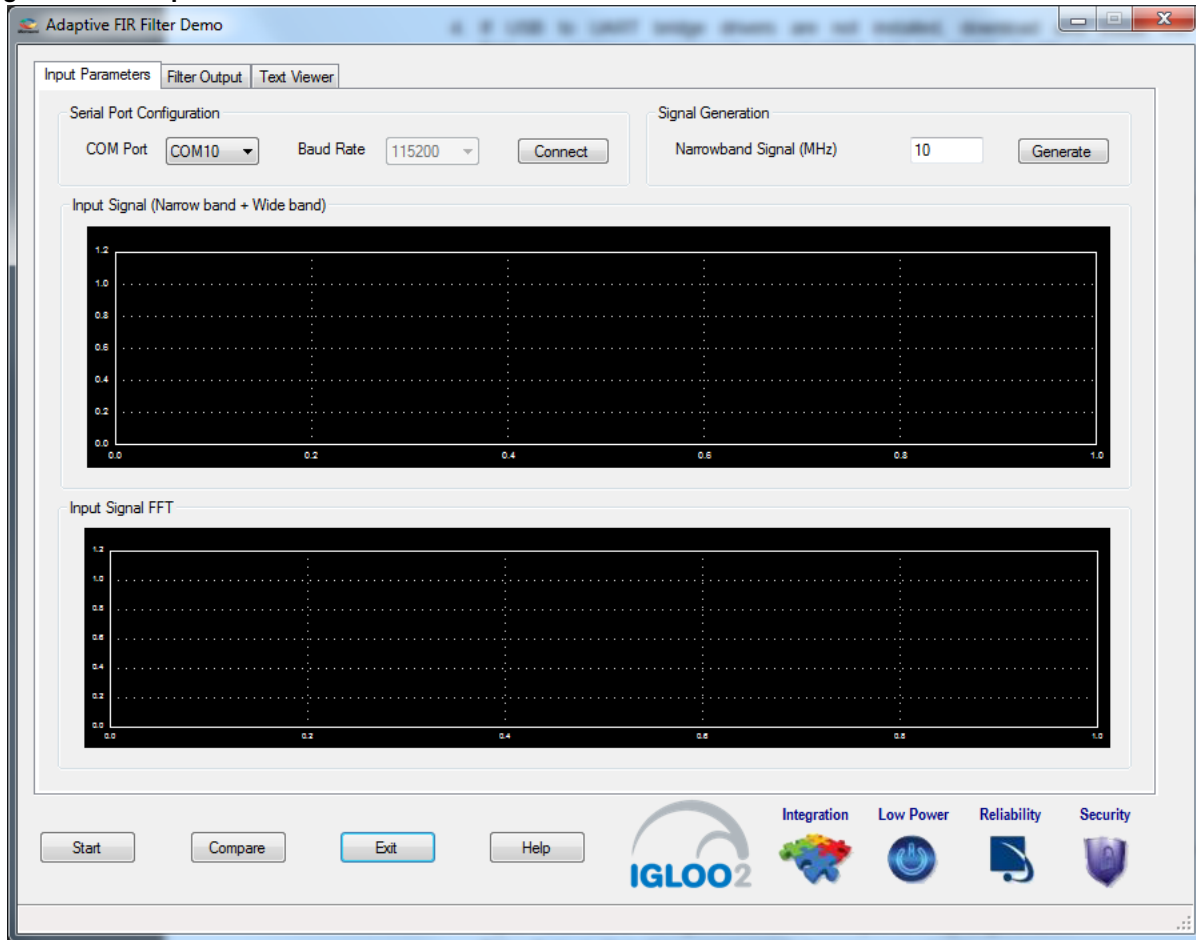
The Adaptive FIR Filter Demo window consists of the following tabs:

- **Input Parameters**: Configures the serial COM port and signal generation.
- **Filter Output**: Plots Error signal and its frequency spectrum.
- **Text viewer**: Shows Input signal, Error signal data values.

Click **Help** for more information on the GUI.

2.6   Running the Demo Design

1. Launch the adaptive FIR filter demo GUI, install and invoke the executable file provided with the design files. (*\\IGLOO2_Adaptive_FIR_Filter_DF\GUI\GL2_Adaptive_FIR_Filter.exe*). The Adaptive FIR Filter Demo window is displayed, see the following figure.

Figure 13 • **Serial Port Configuration**
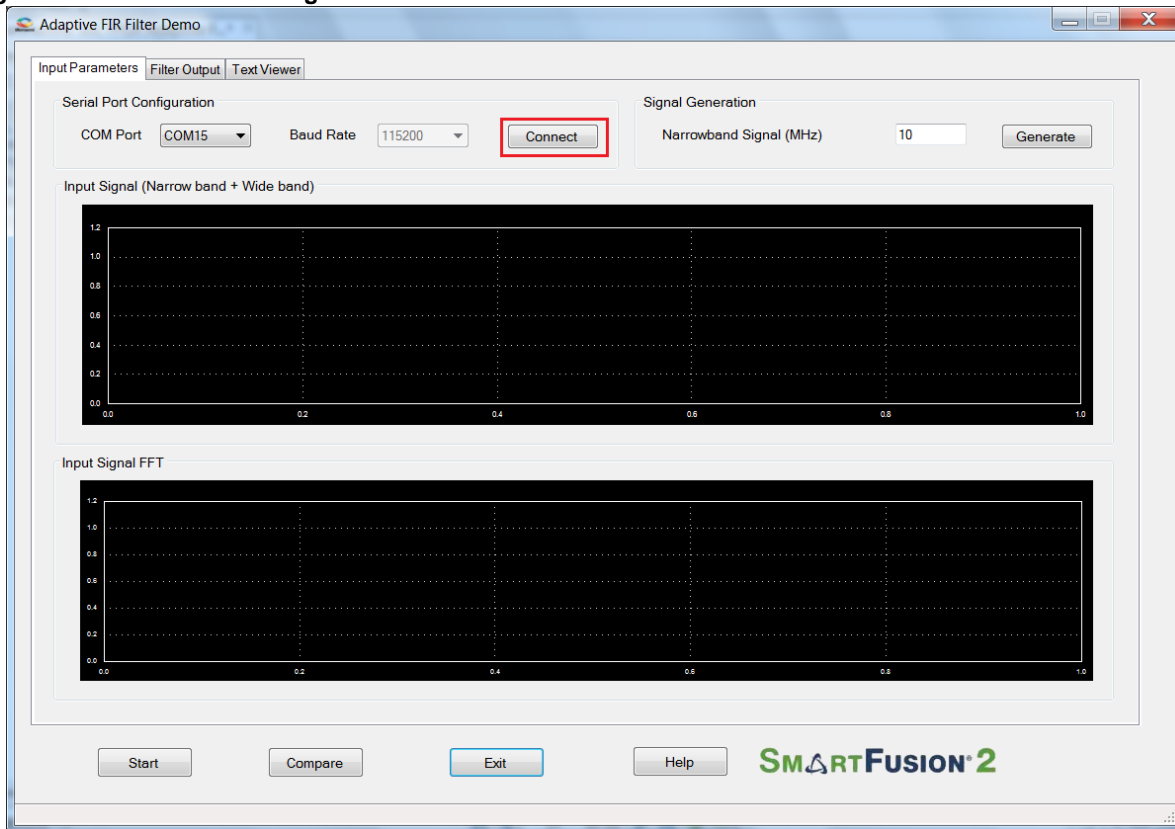
2. **Serial Port Configuration**: The COM port number is automatically detected and baud rate is fixed at 115200. Click **Connect**. See the preceding figure.
3. **Signal Generation**: Enter the narrow band signal frequency as 2 MHz (supported range is 1 MHz and 20 MHz) and click **Generate**. See Figure 14, page 15.

Figure 14 • **Signal Generation**
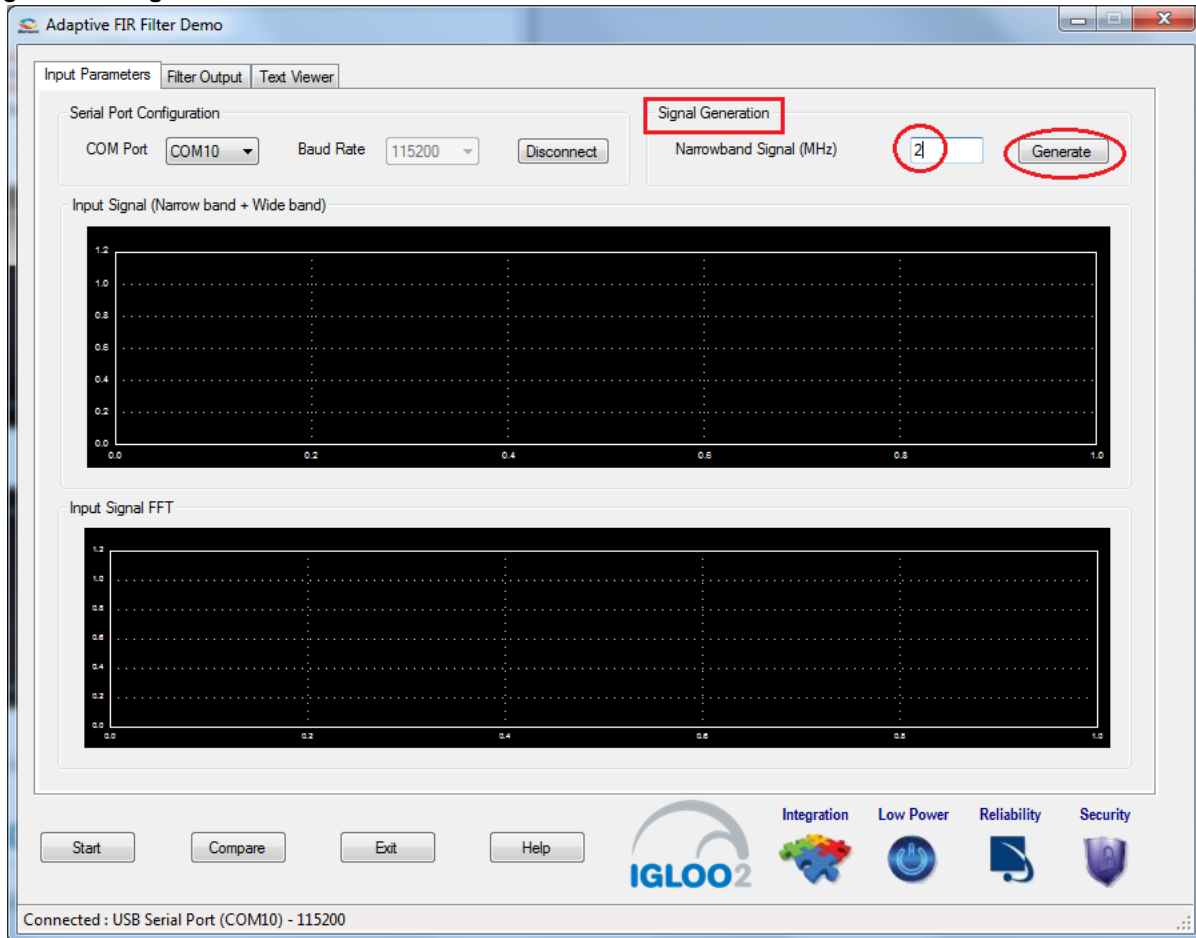
Adaptive FIR filter demo adds the wide band signal (generated inside the Adaptive FIR Filter Demo window) to the narrow band signal component and plots the combined signal (Narrow band and Wide band), FFT spectrum. See Figure 15, page 16.

Figure 15 • **Signal Generation**
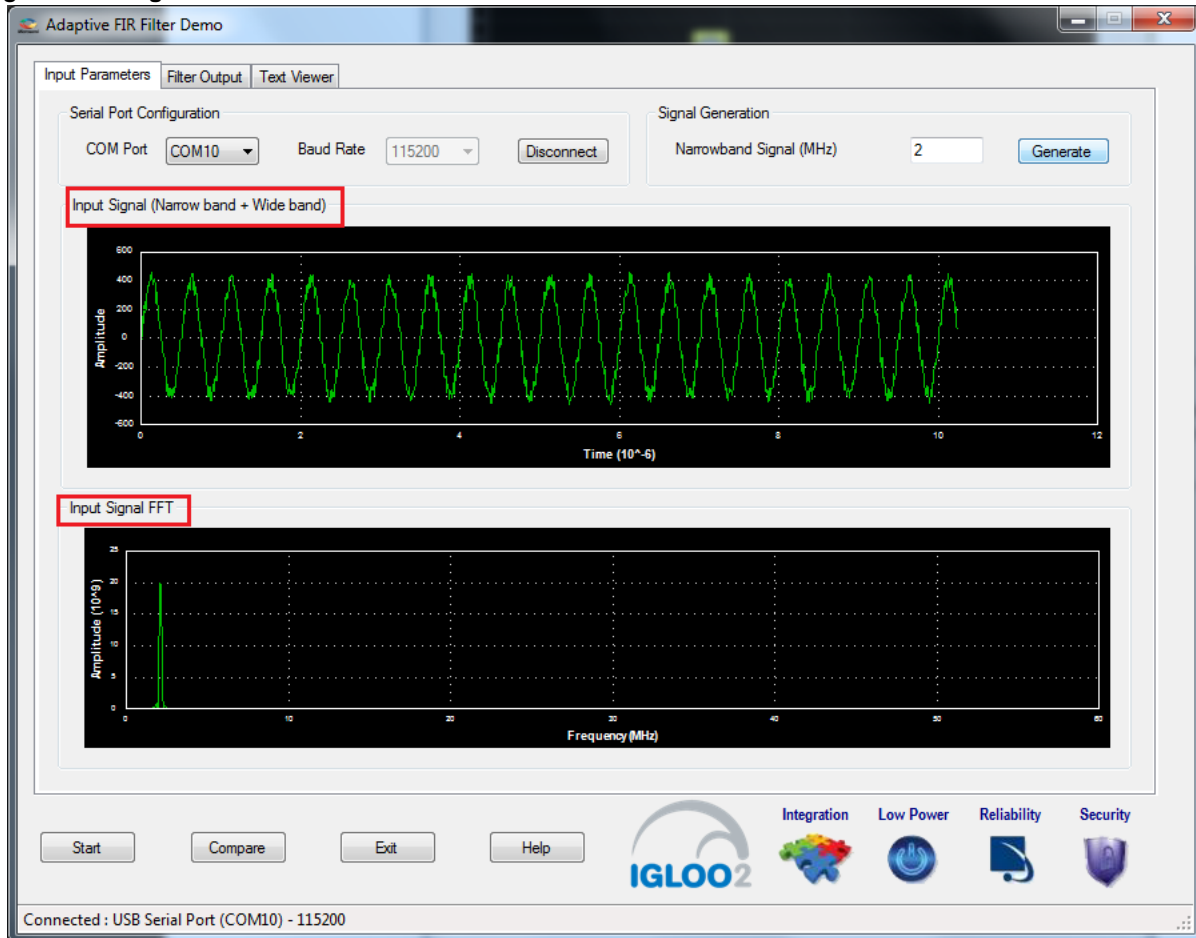
4. Click **Start** to load the input data (1k samples) to the IGLOO2 device for processing the filtering operation, see Figure 16, page 17.

*Figure 16 •* **Adaptive FIR Filter Demo - Start**

After completing the filter operation, the GUI displays the error data and its FFT data from the IGLOO2 device and plots as shown in Figure 17, page 18. The error signal plot shows the suppression of narrow band component from the signal and outputting wide band signals only after the required number of iterations.

Figure 17 • **Error Signal: Time and Frequency Plot**
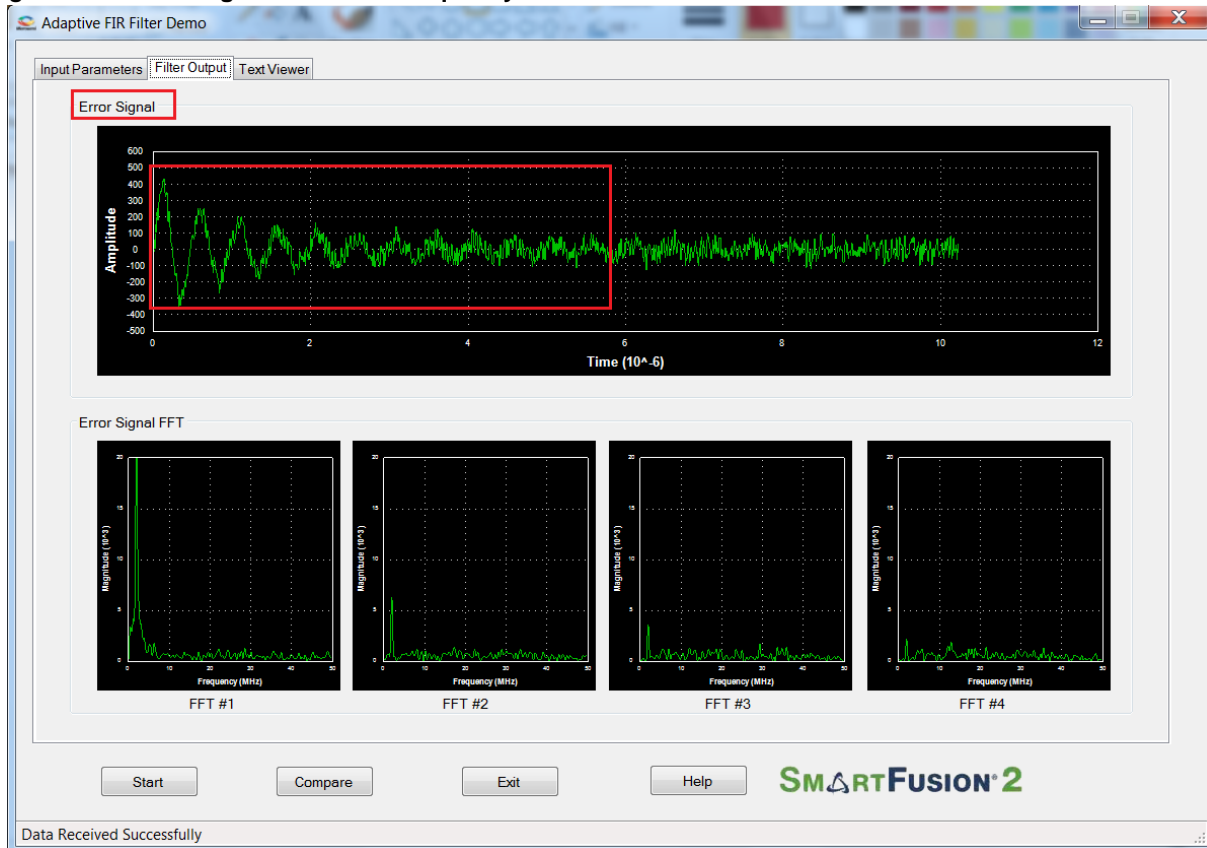
The narrow band signal component is suppressed gradually in the Error signal frequency spectrum. This can be observed in the Error signal FFT plot as shown in Figure 18, page 19.

Figure 18 • **Error Signal FFT: Time and Frequency Plot**
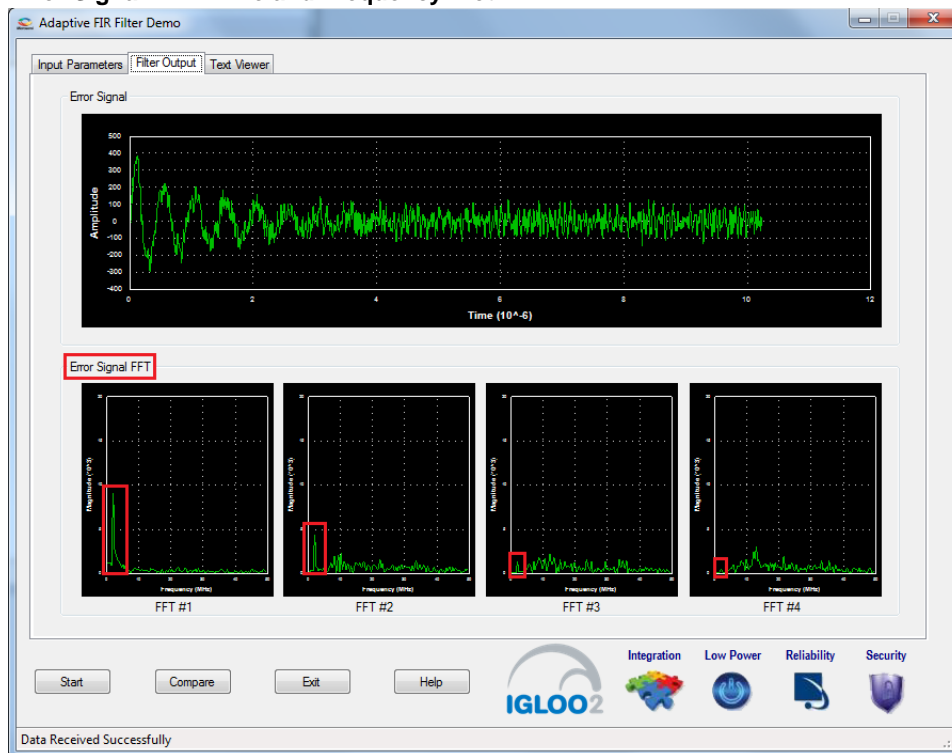
5. Click **Compare** to analyze the Input wide band data with the Output wide band data.

Figure 19 • **Compare Error Signal: Time and Frequency Plot**
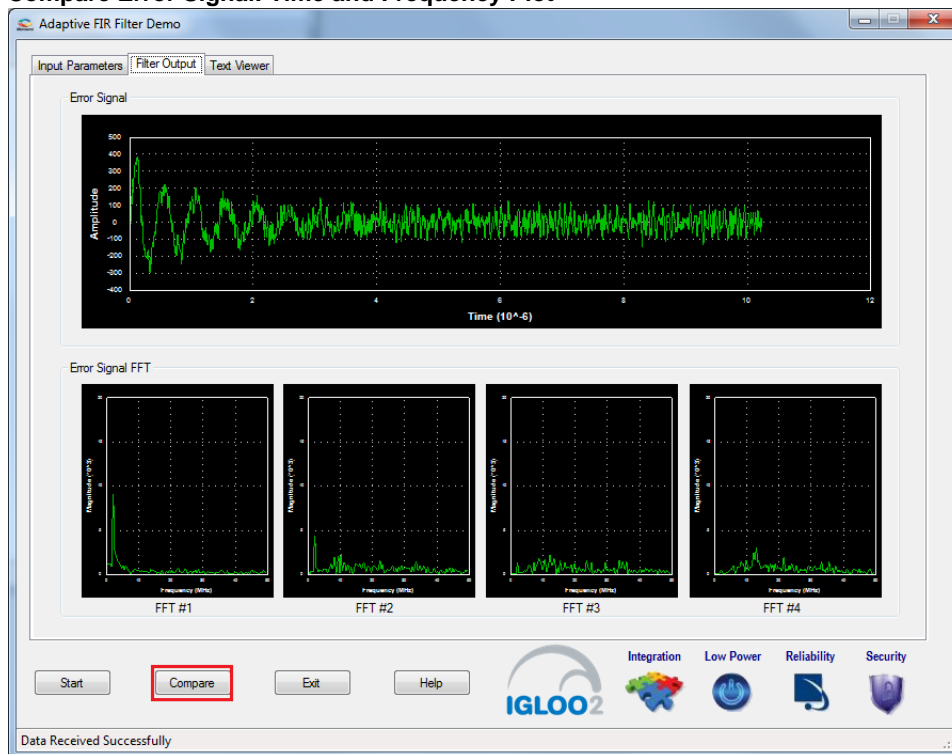
A window displaying the comparison between the Input Wide band and Output Wide band is displayed, see Figure 20, page 20.

*Figure 20 •* **Comparison of Input Wide Band and Output Wide Band**
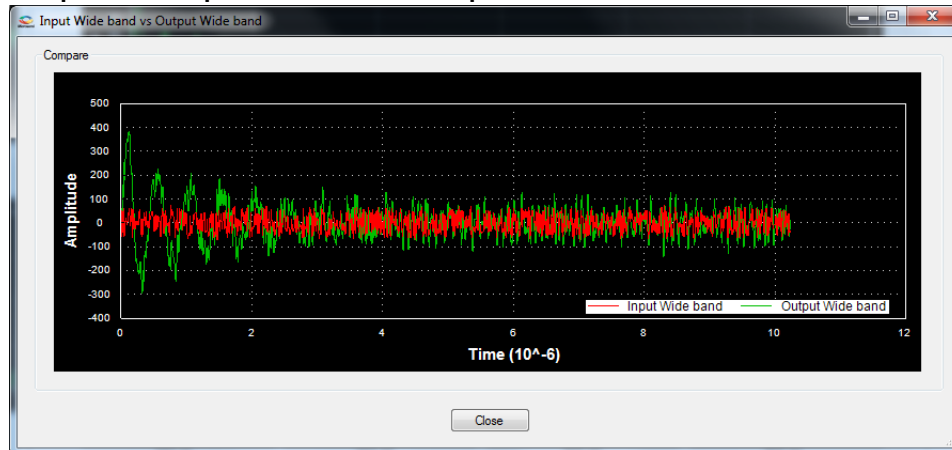
The plot can be zoomed in for comparison, see the following figure.

Figure 21 • **Input Wide Band vs Output Wide Band - Zoomed In**

Compare the Error signal (Output Wide band signal) with the Input Wide band signal, see the following figure. You can see that the narrow band interfering component is eliminated and the wide band signal is preserved in Error signal.

Figure 22 • **Comparison of Input Wide Band and Output Wide Band**
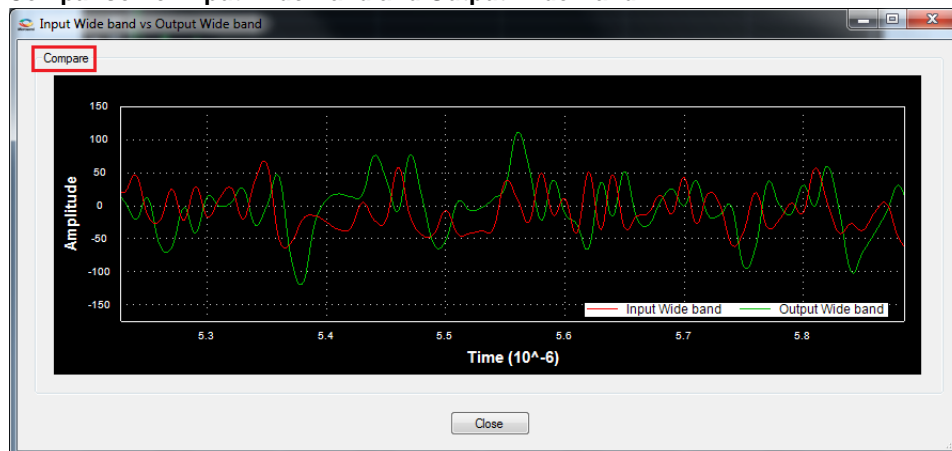
6.   Click **Close**, see Figure 23, page 21.

Figure 23 • **Closing Input Wide Band vs Output Wide Band Window**

7. You can copy, save, export and customize page and configure print setup the Error Signal plot. Right-click **Error Signal** plot.
8. From the context sensitive pop up select the required option.

It shows the different options as shown in the following figure. The data can be copied, saved, and exported to CSV plot for analysis purpose. Page setup, print, show point values, Zoom, and set scale are set to default.

*Figure 24 •* **Error Signal - GUI Options**

9. The input signal and error signal values can be viewed in the **Text Viewer** tab. Click the **Text Viewer** tab and then click the corresponding **View** shown in Figure 25, page 22.

Figure 25 • **Text Viewer**
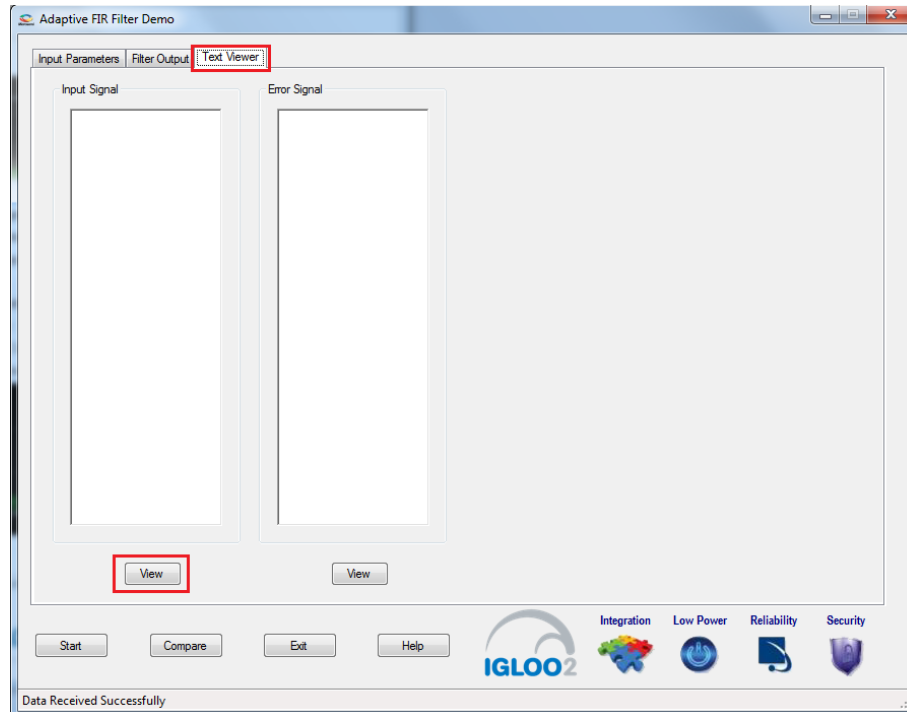
The following figure shows the Text Viewer tab showing the Input Signal values.

Figure 26 • **Text Viewer: Input Signal Values**
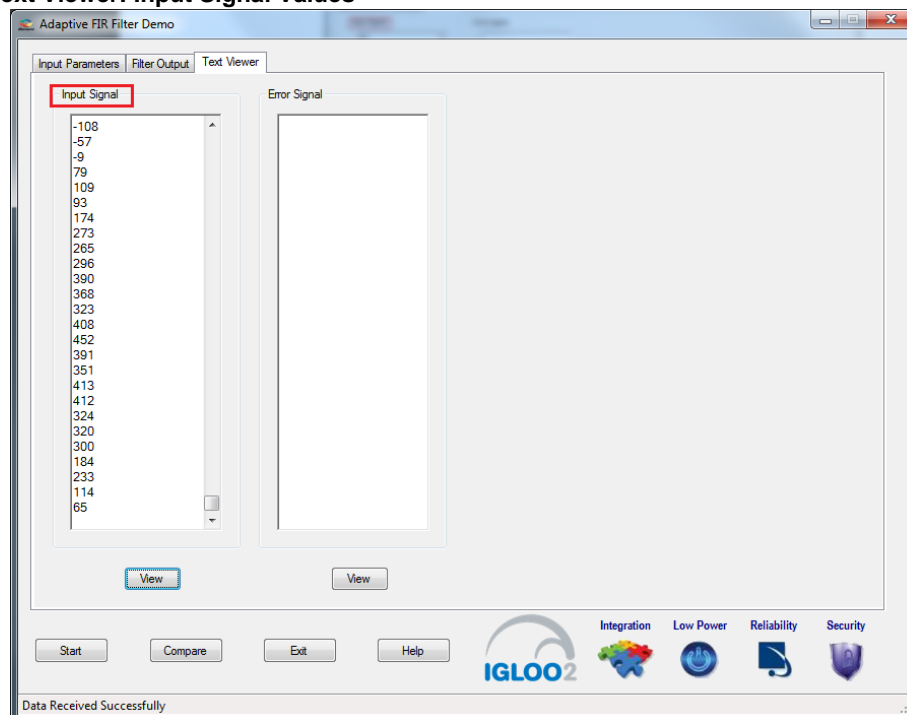
10. To save the Input Signal as a text file, right-click the Input Signal window. The Input Signal window displays different options as shown in the following figure.
11. Click **Save**. Select **OK** to save the text file.

Figure 27 • **Text Viewer - Input Signal Save Option**
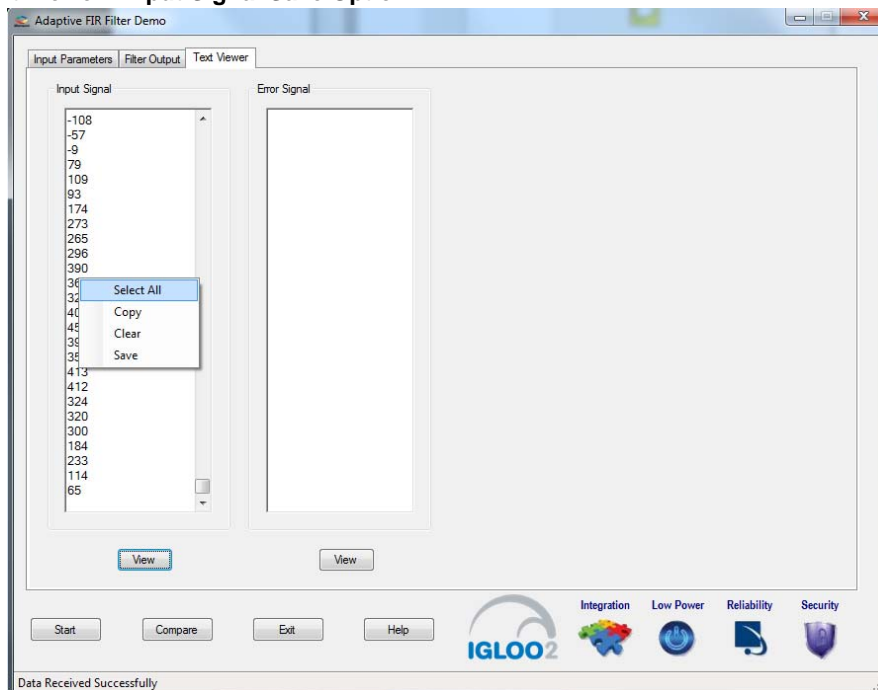
12. Click **Exit** to stop the demo, see the following figure.

Figure 28 • **Exit Demo**
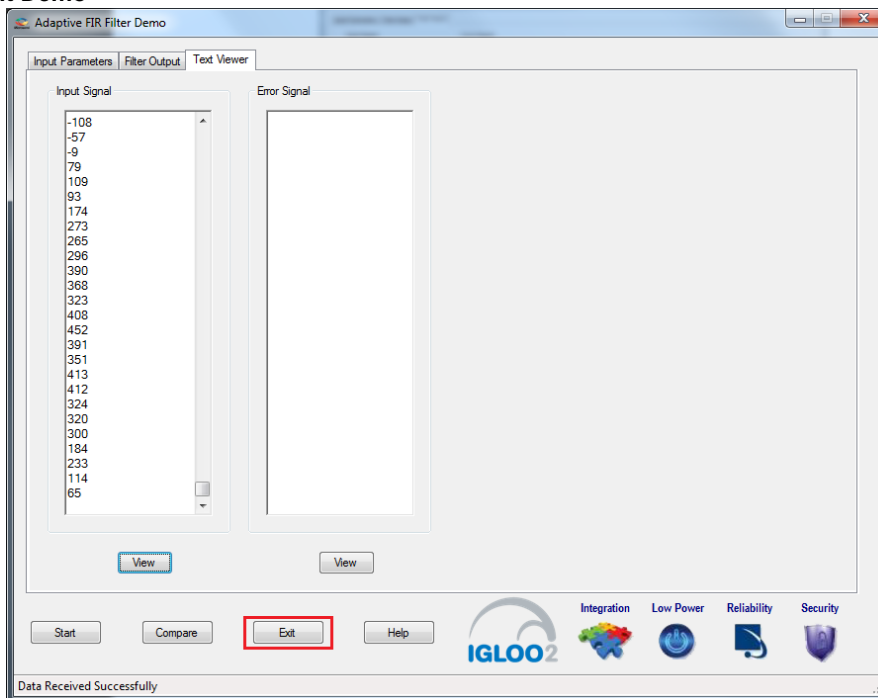

2.7 Conclusion

This demo provides information about the features of the IGLOO2 device including math blocks and how to use Microsemi IPs (CoreFIR and CoreFFT) for narrow band interference cancellation application using Adaptive FIR filters. This Adaptive FIR filter based demo is easy to use and provides several options to understand and implement DSP filters on the IGLOO2 device.

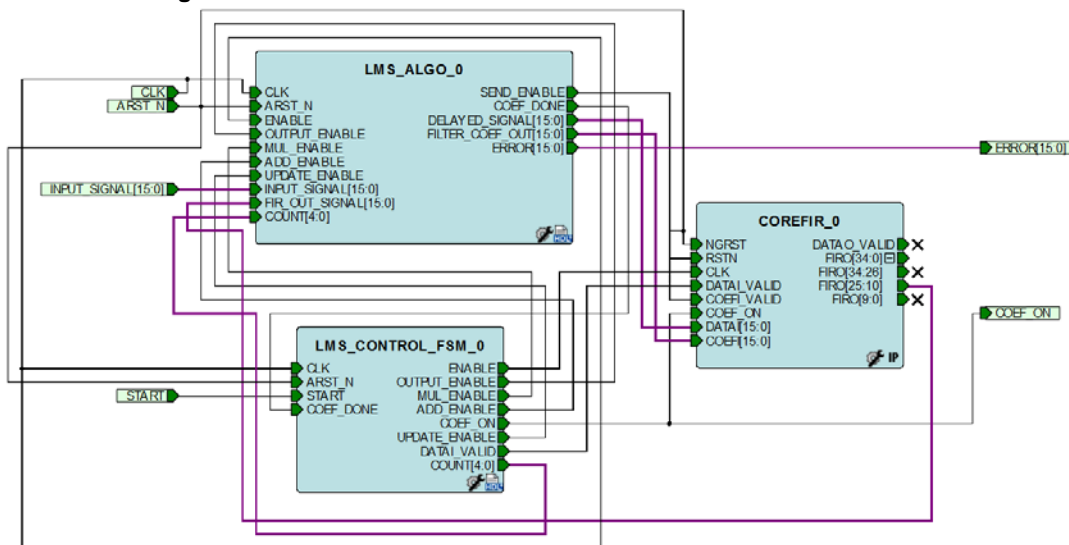# 3 Appendix: SmartDesign Implementation

Adaptive FIR filter SmartDesign is shown in the following figure.

Figure 29 • **Adaptive FIR Filter SmartDesign**
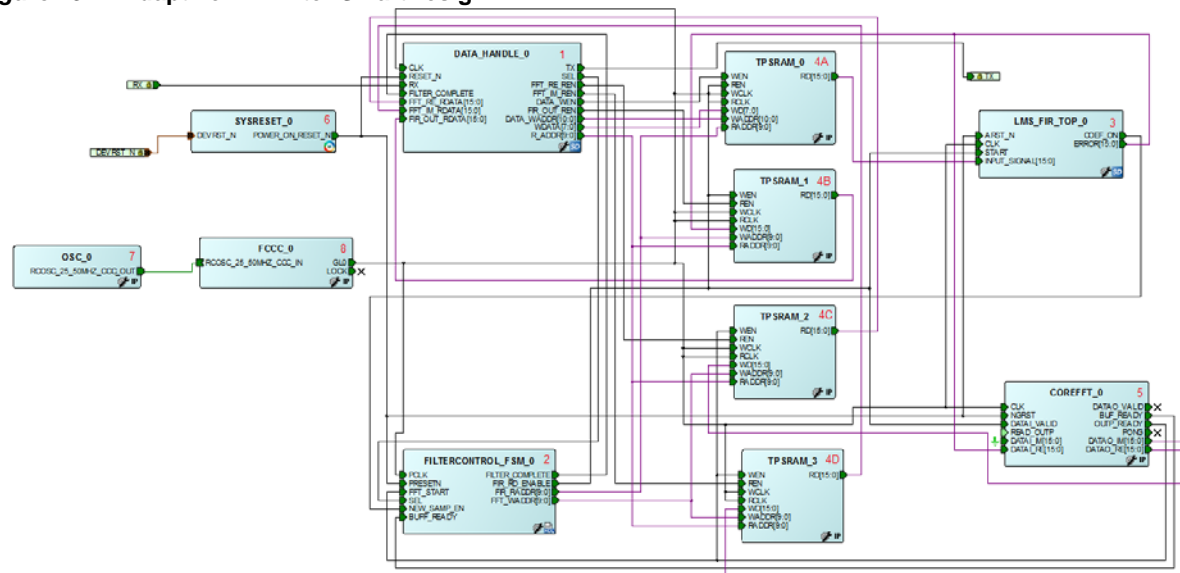
SmartDesign LMS_FIR_TOP is shown in the following figure.

Figure 30 • **SmartDesign LMS_FIR_TOP**

The following table shows SmartDesign blocks in Adaptive FIR filter.

Table 4 •   **Adaptive FIR Filter Demo SmartDesign Blocks and Description**

S.No	Block Name	Description
1	DATAHANDLE_0	Handles communication between the host PC and IGLOO2 Evaluation Kit board.
2	FILTERCONTROL_FSM_0	Control logic to generate the control signals for FIR and FFT operations.
3	LMS_FIR_TOP	SmartDesign.
4	INPUT_Buffer	FIR input signal data buffer.
	OUTPUT_Buffer	FIR output signal buffer.
	FFT_Im_Buffer	FFT output imaginary data buffer.
	FFT_Re_Buffer	FFT output real data buffer.
5	COREFFT_0	COREFFT IP.
6	SYSRESET_0	Reset IP.
7	OSC_0	Oscillator IP.
8	FCCC_0	Clock conditioning circuit IP.

The following table shows SmartDesign blocks in LMS_FIR_TOP.

Table 5 •   **LMS_FIR_TOP SmartDesign Blocks and Description**

S.No	Block Name	Description
1	LMS_ALGO	LMS algorithm implemented in the RTL to compute error, correction factor, and filter coefficients.
2	LMS_CONTROL_FSM	FSM implemented in the RTL to control LMS_ALGO block.
3	COREFIR	COREFIR IP.

4   Appendix: Resource Usage Summary

The following table shows Adaptive FIR filter demo resource usage summary.

Device: IGLOO2 device

Die: M2GL010

Package: 484 FBGA

Table 6 •   **Adaptive FIR Filter Demo Resource Usage Summary**

Type	Used	Total	Used vs Total in Percentage (%)
4LUT	3135	12084	25.94
DFF	2899	12084	23.99
RAM1K×18	11	21	52.38
MACC	13	22	59.09

The following table shows MACC blocks usage summary.

Table 7 •   **MACC Blocks Usage Summary**

CoreFIR	CoreFFT	LMS_ALGO	Total
8	04	1	13

The following table shows RAM1Kx18 blocks usage summary.

Table 8 •   **RAM1Kx18 Blocks Usage Summary**

CoreFIR	CoreFFT	Fabric Buffers	Total
0	7	4	11