


Introduction

This application note describes the tool flow for designing a digital intermediate frequency (IF) modem using the DSP Builder Advanced Blockset.

DSP Builder is a digital signal processing (DSP) development tool interface for designs developed in the Simulink environment through to FPGA implementation. DSP Builder contains two library blocksets:

- The standard blockset has cycle accurate behavior and is suitable for managing multiple clock domains when interfacing to external logic.
- The advanced blockset is ideal for designs with continuous data streams such as designs in RF cards. An IF modem design uses chains of filter and is an ideal application for the advanced blockset.

 For information about the tool flow for the DSP Builder standard blockset, refer to [AN442: Tool Flow for Design of Digital IF for Wireless Systems](#).

The target applications for the DSP Builder Advanced Blockset includes:

- High-performance finite impulse response (FIR) and infinite impulse response (IIR) filters.
- Digital up conversion: Code Division Multiple Access (CDMA), Wideband CDMA (W-CDMA), Worldwide Interoperability for Microwave Access (WiMAX), Long Term Evolution (LTE) wireless technologies, software-defined radio, high end broadcast, and so on.
- Digital down conversion: CDMA, W-CDMA, WiMAX, LTE wireless technologies, radar and software-defined radio, medical imaging, and so on.
- Fast Fourier transform (FFT) and inverse fast Fourier transform iFFT: Radix2/4 engine, and custom designs with non-standard lengths.

Background on Digital IF Modem Design

To relax the direct analog modulation and demodulation specifications in radio frequency (RF), baseband signals are converted to an intermediate frequency (IF) in the digital domain followed by analog processing and vice versa.

Digital IF modem designs fulfil an intermediate role between baseband and RF. It is an essential part of the RF card solutions in wireless standards such as WiMAX, W-CDMA, and LTE. With different wireless technologies evolving and shorter time to market, it is important to build a system with flexibility for future upgrade and maintenance.

An IF modem comprises of a digital upconverter (DUC) in the transmitter and a digital downconverter (DDC) in the receiver.

In a DUC, the complex baseband signals are interpolated to IF sampling rate and modulated up to selected IF carrier frequencies ranging from 0 Hz to $(\frac{1}{2}\text{sample rate} - \text{baseband bandwidth})$. Sometimes the IF carrier frequency is chosen as one quarter of the sampling rate to further reduce hardware multiplier resource utilization.

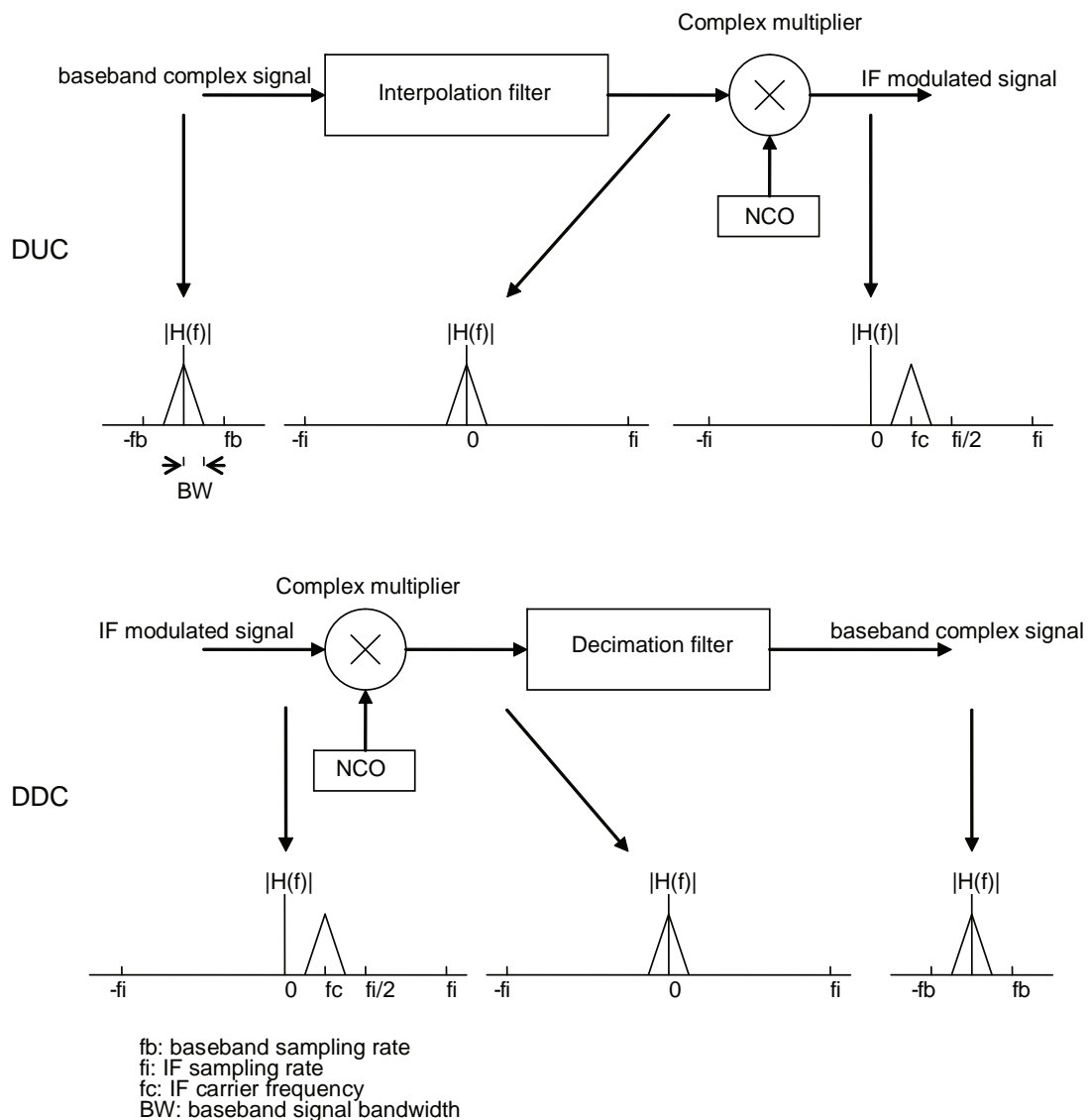
For W-CDMA, you can choose to have one or more carriers transmitted using one antenna. In the example designs described by this application note, the modulated up-converted signals are summed together before output to antenna.

For WiMAX, there is no summation because there is only one carrier frequency. In the example designs, modulated up-converted signals are output individually to antenna.

In a DDC, the real IF signals are demodulated from selected carrier frequencies, and decimated to base band sampling rate.

Figure 1 shows the generic block diagram of DUC and DDC in a wireless system and the corresponding spectrum at each stage.

Figure 1. Generic Block Diagram of DUC and DDC in a Wireless System



DSP Builder Advanced Blockset Tool Flow

The DSP Builder advanced blockset is based on a high-level synthesis technology that optimizes the untimed netlist into low level, pipelined hardware targeted to your chosen FPGA device and chosen clock rate. The advanced blockset provides constraint driven design methodology through system level parameters. You specify the desired f_{\max} , the tool then inserts the right amount of pipeline and maintains the data path algorithm accuracy.

For system engineers, the advanced blockset introduces them to FPGA design without having to immediately comprehend all of the trade-offs associated with other FPGA design flows. For hardware engineers, it allows designs to be completed in an abstract format, eliminating the need for timing closure through auto-pipelining and generation of any control logic.

For the design of digital IF modems, the DSP Builder Advanced Blockset offers a high level design entry point, which allows you to explore the design space. The provided IP reduces the design time for commonly used functional blocks.

The following sections explain the tool flow of the IF modem design.

Base Blocks

The top-level of a DSP Builder Advanced Blockset design is a testbench and must include `Control` and `Signals` Blocks from Base Blocks library. The functional subsystem containing a `Device` block marks the top level of the FPGA device and specifies the target device used for the generated hardware.

Unified Interface

The IF modem design is built using the blocks from the `ModelIP` and `ModelPrim` libraries in the DSP Builder Advanced Blockset. The `ModelIP` library includes a set of multichannel, multi-rate filters in the `Filter` library, plus mixers and a numerically controlled oscillator (NCO) in the `Waveform Synthesis` library. The `ModelPrim` library provides functional primitive blocks that are used to build custom logic.

`ModelIP` and `ModelPrim` blocks can be connected together using a standard `<data, valid, channel>` protocol. Data is only valid when the `valid` signal is asserted high and the `channel` signal carries an 8-bit integer channel identifier. This channel identifier is preserved through the data path so that data can be easily tracked and decoded.

[Figure 2 on page 4](#) shows the port interface for `ModelIP` and `ModelPrim` blocks.

[Figure 2](#) also shows that the data types of filter blocks are different from waveform synthesis blocks. Filter blocks are intended for general purpose and process real data only. The complex mixer blocks accept complex input data and sinusoids to generate complex output data while the real mixer takes real data with complex sinusoids to output complex data. Because of this difference between filter and mixer blocks, an adapter is required to convert the data format.



More detail about the adapter logic is given in [“Multichannel TDM Operation” on page 5](#).

Figure 2. Port Interface of ModelIP and ModelPrim Blocks

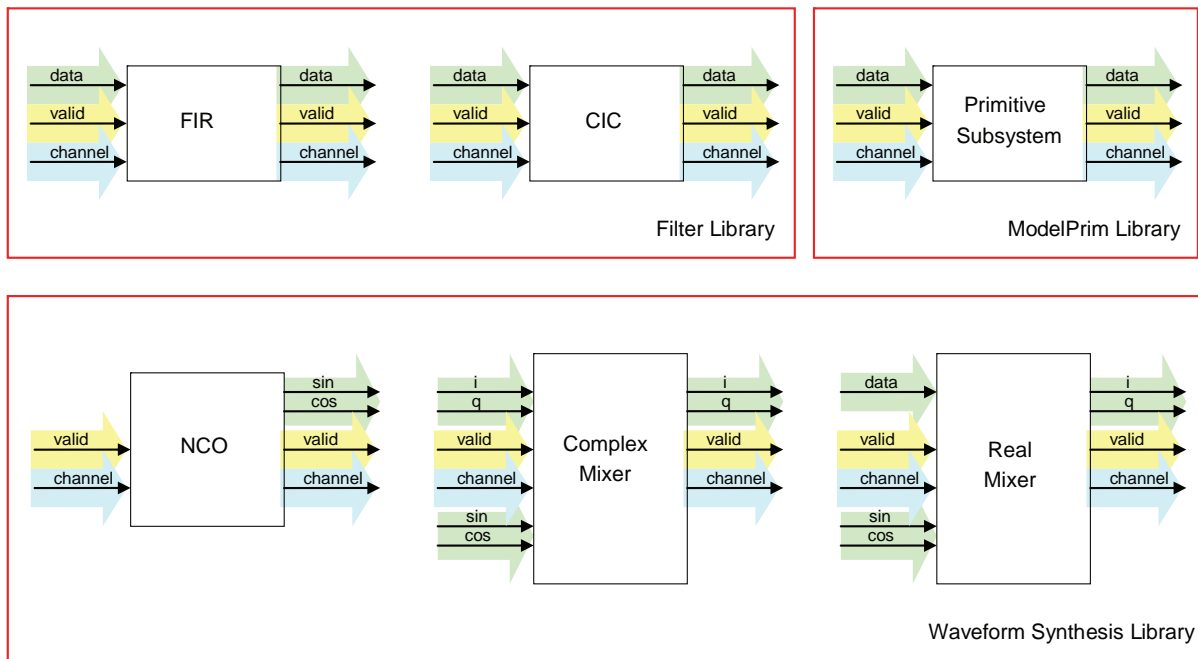


Figure 3 shows the connection between the filters and complex mixer in the DUC.

Figure 3. Connecting a FIR or CIC Filter to a Complex Mixer Using a Splitter

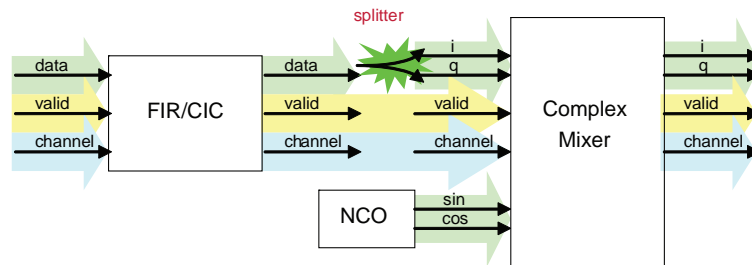
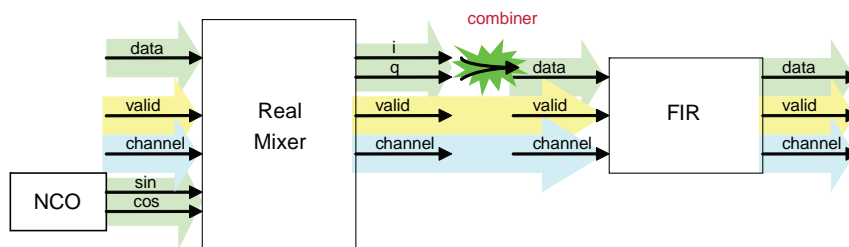


Figure 4 show the connection between the real mixer and filters in the DDC.

Figure 4. Connecting a Real Mixer to a FIR Filter Using a Combiner



Time-Division Multiplexing (TDM) Support

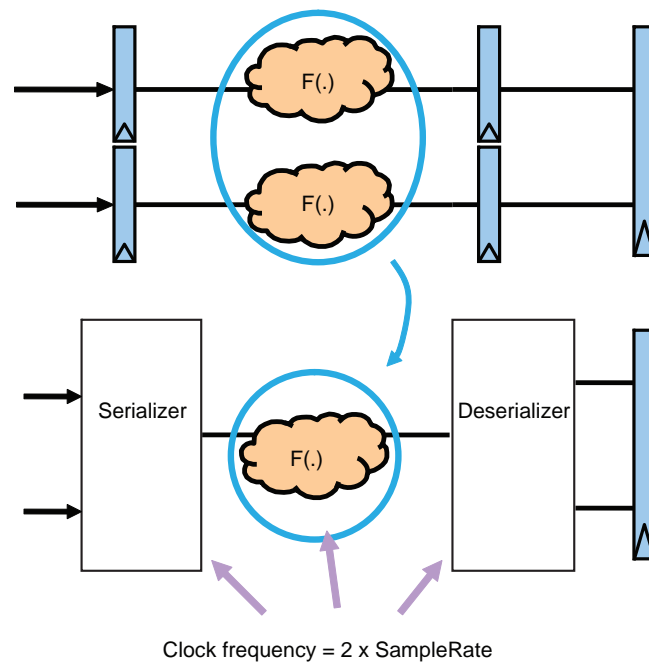
The sample rate is the rate at which real data is clocked through the system at any point. Usually the maximum clock frequency in a FPGA is much higher than the data rate, which allows efficient resource usage. ModelIP blocks offer TDM (also known as folding) support based on the specified clock frequency, sample rate, and rate change factor in the multirate filters.

The TDM technique works by clocking the TDM block n -times faster than the sample rate. This technique re-uses the hardware and performs the same functionality up to n times where n is the ratio of clock frequency to sample rate.

Figure 5 shows an example with TDM factor 2 to implement the top data path with two functional blocks using one functional block on the bottom clocked two times faster.

The ratio of system clock frequency to sample rate determines the amount of resource, which translates to how you can pack more channels with a fixed amount of resources or use fewer resources with a fixed number of channels. This time sharing reduces the overall cost of implementation.

Figure 5. Example of Time-Division Multiplexing With TDM Factor = 2



All the blocks provided in the ModelIP libraries support these folding capabilities based on the system parameters.

Multichannel TDM Operation

In a multi-rate environment, the sample rate varies along the filter chain. All ModelIP blocks support multichannel TDM operation. Instead of duplicating a single channel design multiple times to support multichannels, the multichannel input and output samples are multiplexed into the ModelIP components when the clock frequency is higher than sample rate per channel.

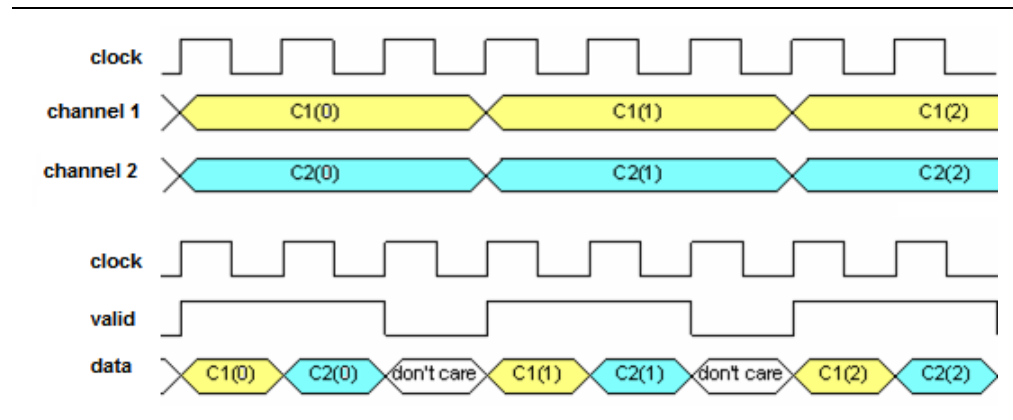
When the aggregated data rate exceeds the clock frequency, multiple hardware components are required to handle the excess data rate. However, the high level design in Simulink remains the same while the connecting wires expand to multiple dimensions to maintain the cleanness and scalability of the top level design. The multiple instantiation of design building blocks is handled by the tool during RTL generation.

The parameters in the demonstration design are specified using the following variables:

- *ClockRate* — system clock frequency (MHz).
- *SampleRate* — input sample rate per channel (MSPS).
- *ChanCount* — number of channels.
- $Period = \text{floor}(ClockRate / SampleRate)$ — available timeslots per wire (sometimes interchangeably referred as TDM factor or folding factor).
- $ChanWireCount = \text{ceil}(ChanCount / Period)$ — number of wires required to carry all the channels. (That is, the number of times that the same module needs to be duplicated to support all the channels.)
- $ChanCycleCount = \text{ceil}(ChanCount / ChanWireCount)$ — number of carried channels per wire. (This value cannot exceed *Period*. When it is less than *Period*, some time slots on a wire are not used.)

In [Figure 6](#), the ratio of clock frequency to sample rate per channel is 3, which means that three channels can fold into one wire. The ratio between *ClockRate* and *SampleRate* is more significant than the absolute values when demonstrating the folding feature. The timing diagram shows that the valid signal is high for two cycles and low for one cycle to accommodate the two channels that are available.

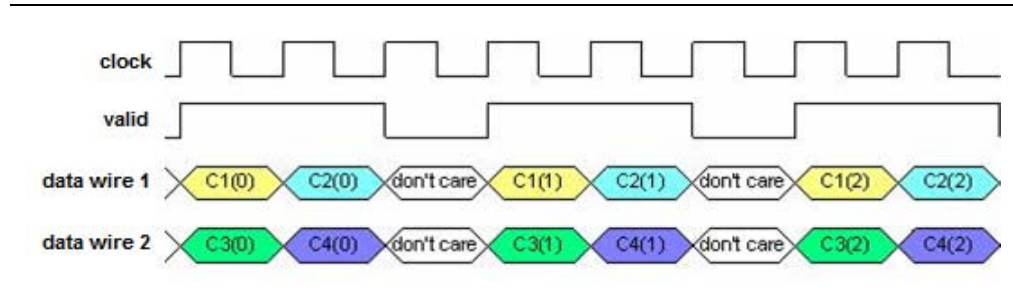
Figure 6. Timing diagram with $Period = 3$, $ChanCount = 2$



When the number of channels exceeds the available timeslots, multiple chains are required to support the aggregated sample rate. In that case, the ModelIP block internally duplicates to multiple instances in parallel and wires one element of the vector up to each instance.

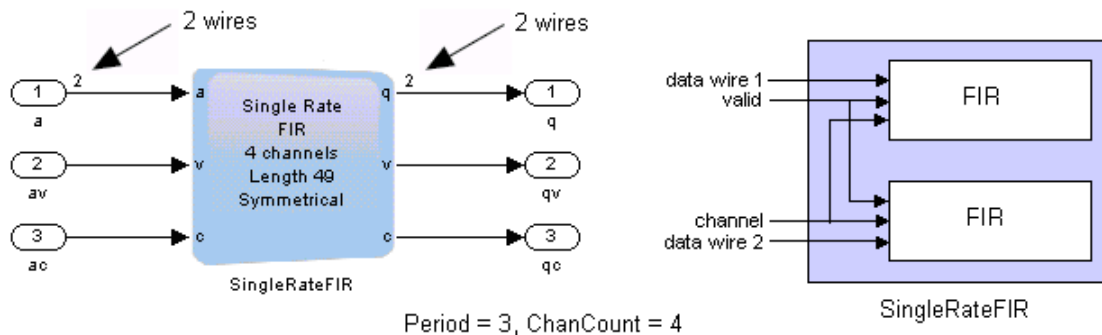
Figure 7 shows the timing diagram for a four channel system with $Period = 3$. In this example $ChanWireCount = 2$ and $ChanCycleCount = 2$.

Figure 7. Timing diagram with $Period = 3$, $ChanCount = 4$



As shown in Figure 8, the top level still explicitly shows one instance in Simulink. However, the data input and output wires from the ModelIP blocks are shown as a vector with multiple dimension. Multiples wires are required to accommodate all the channels and the Simulink model uses a vector of width 2.


Figure 8. Simulink and Hardware Representations of a Single Rate FIR Filter



In this example, the clock rate is three times the sample rate per channel and there are four channels. Two wires are required to accommodate the four channels and on each wire the cycle count equals two. With the unified interface protocol, the tool decodes the data from the channel identifier along with the number of wires as shown as cycle index vector and wire index vector.

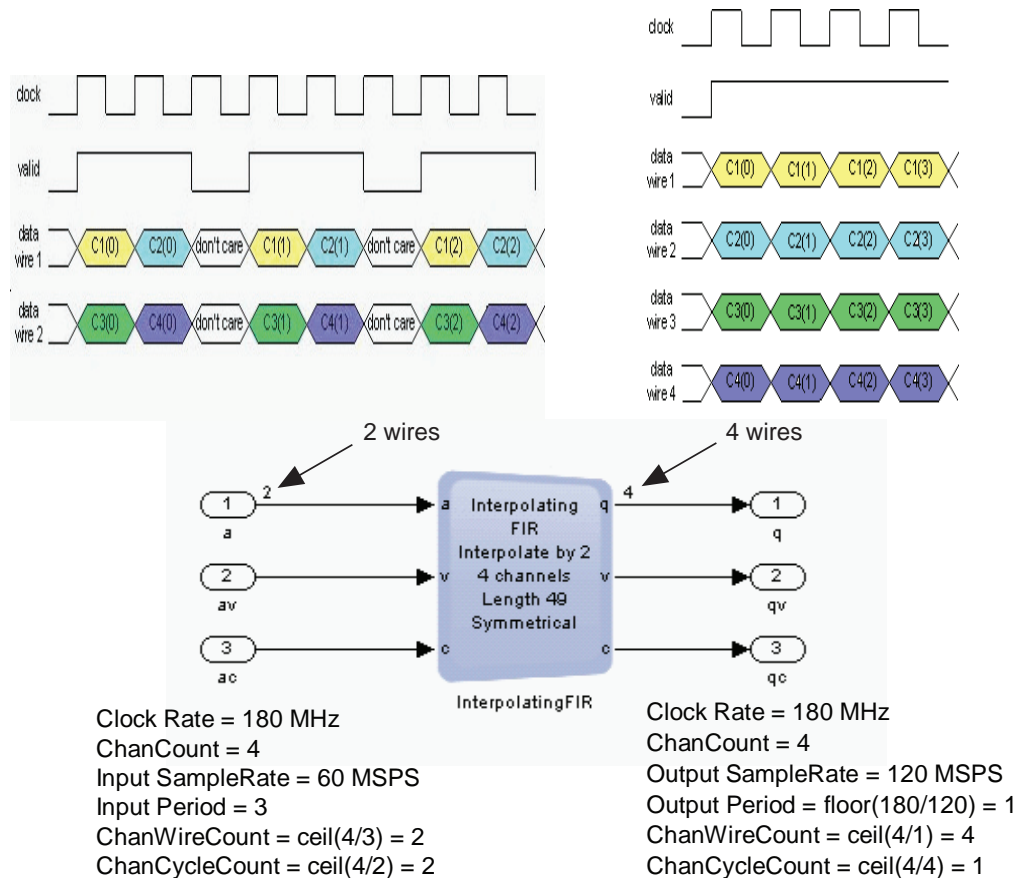
The channels are evenly distributed among the wires rather than filling the first then the second. Thus, the channels are arranged on the top wire left to the $ChanCycleCount$ and then on bottom wire left to $ChanCycleCount$. The valid signal for the third timeslot is low and the data in these timeslots is ignored for both wires.

This method is better than a traditional tool flow because there is no need to manually instantiate multiple ModelIP blocks.

 To display the $ChanWireCount$ in Simulink, choose **Format -> Port/Signal Displays** and turn on **Signal Dimensions**.

When there is a sample rate change involved, the Simulink design does not change while the underlying hardware scheduling and implementation is automatically performed. Figure 9 shows the input and output timing diagrams for an interpolating by two FIR filter. The input data wire is a vector of two elements while the output data wire is a vector of four elements. As the scenario gets more complex, many of the background details are invisible. It can then be useful to annotate the parameter values as shown in Figure 9 to keep track of each stage.

Figure 9. Timing Diagram for an Interpolating FIR Filter with Period = 3, ChanCount = 4



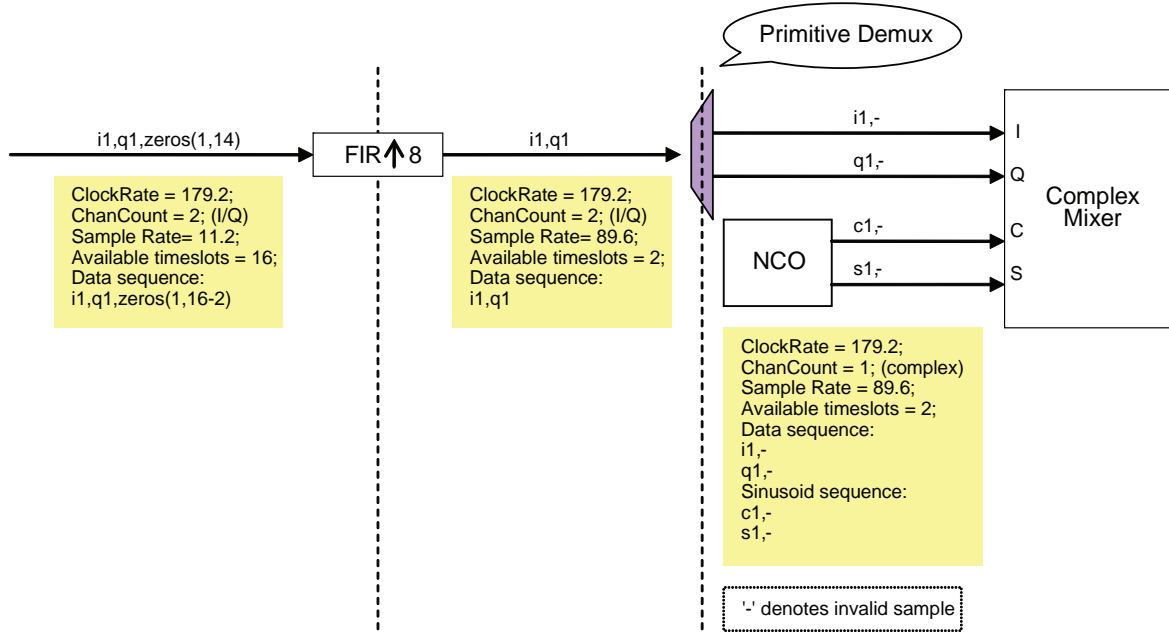
The advantage of the vectorized support is to provide productivity gains for design exploration through system constraints without having to re-architect the design.

Multiplexed IQ Channels Out of Filter Chain

When the clock rate is higher or equal to the aggregated sample rate at the end of a filter chain, the output I and Q channels from the FIR filter are time multiplexed. A demultiplexer is required to separate the I stream and Q stream into two separate wires and control logic is required to update the channel and valid signals.

The demultiplexer logic can be implemented by components from the ModelPrim library. For example, the data stream of $i1, q1$ with channel count from 0,1 are converted to two data stream of $i1,-$ and $q1,-$ with channel count as 0,-. The valid signal is valid half of the time as well. Figure 10 on page 9 shows the connections from the FIR multiplexed IQ channels to the complex mixer.

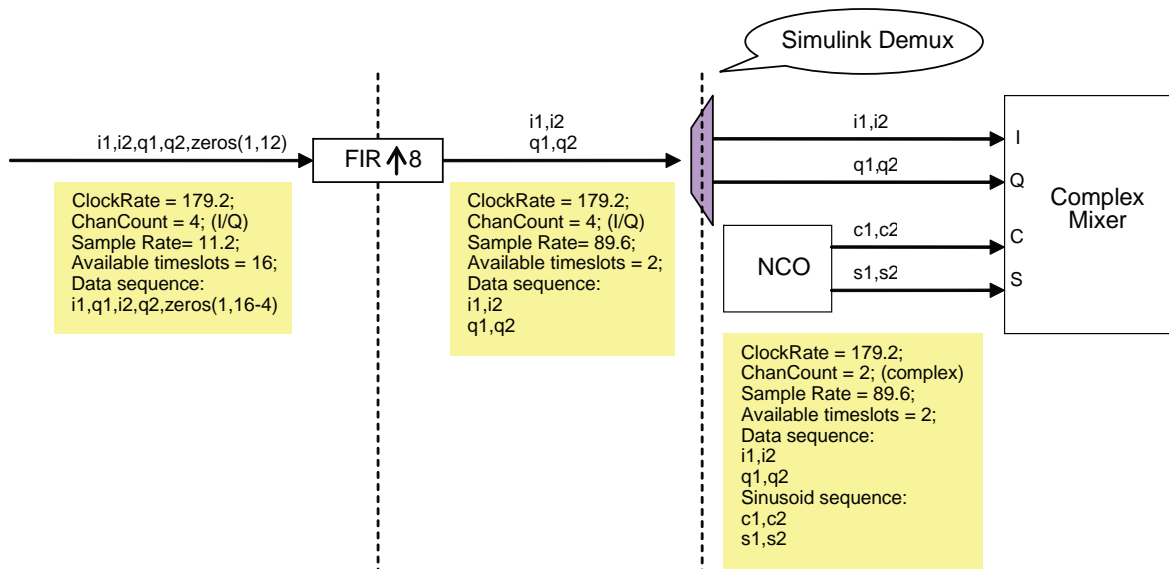
Figure 10. Connection From FIR Multiplexed IQ Channels to Complex Mixer




Parallel IQ Channels Out of Filter Chain

In some cases, the aggregated multi-channel sample rate at the FIR filter output exceeds the clock frequency and the I and Q channels are carried in a wire of separate dimensions. A bus splitter is sufficient to separate this wire into two wires connected to the complex mixer. **Figure 11** shows the connections from the FIR parallel IQ channels to a Complex Mixer. To align the IQ signals at the Filter output, the input must have all multiplexed I channels followed by the Q channels as shown in the example.

Figure 11. Connection From FIR Parallel IQ Channels to the Complex Mixer



 You can implement a bus splitter using the Demux block from the Signal Routing library in the Simulink blockset. (This block is synthesizable when used with the DSP Builder Advanced Blockset.)

ModelIP Portfolio

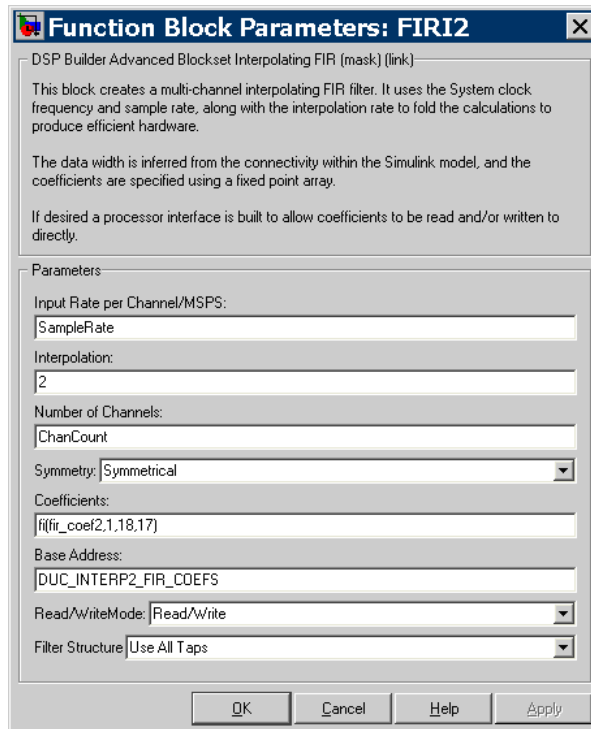
The ModelIP blocks offer highly efficient intellectual property (IP) blocks to cut the design time on building commonly used functionality. The ModelIP blocks can be categorized into the Filter library and Waveform Synthesis libraries.

FIR Filter

The Filter library contains a number of decimating and interpolating cascaded integrator comb (CIC) and finite impulse response (FIR) filters including single rate, multi-rate and fractional rate FIR filters. Multi-rate filters are essential to the up and down conversion tasks required in modern radio systems.

Figure 12 shows the function block parameters GUI for an interpolating FIR.

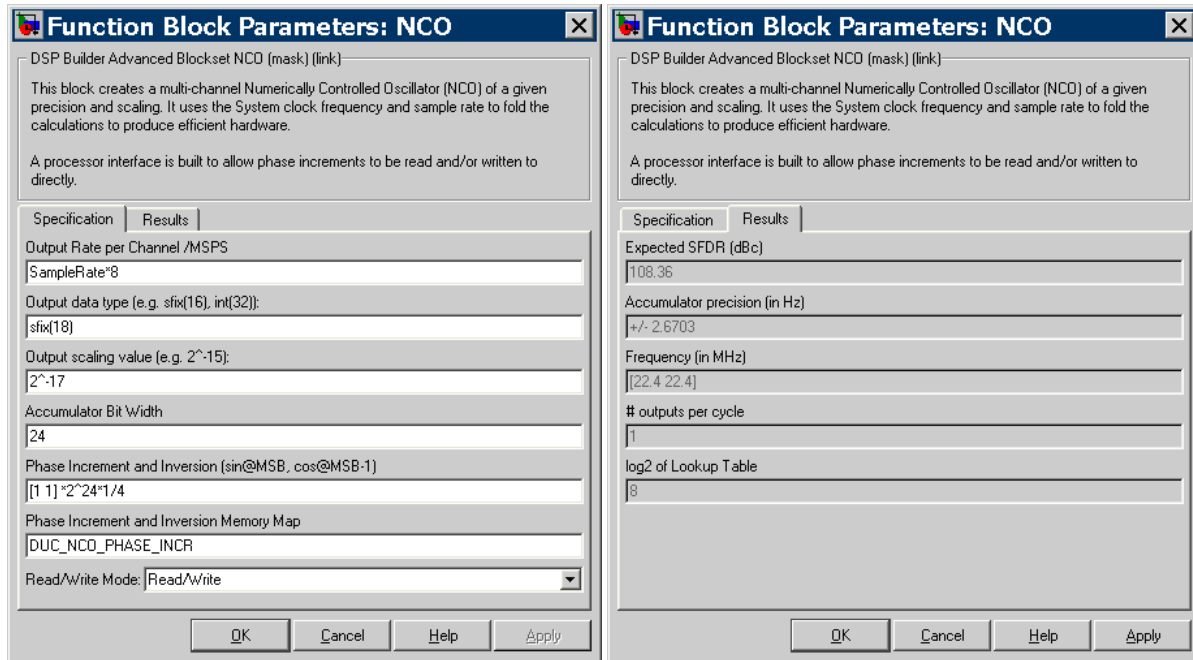
Figure 12. Function Block Parameters GUI for Interpolating FIR.



NCO

Numerically controlled oscillators (NCO) are efficient means of generating sinusoidal signals. When the IF carrier frequency can be chosen at 1/4 of the IF sampling frequency, the values of sine and cosine outputs are -1,0, and 1. The mixer can be implemented with just adders to save multiplier usage. Figure 13 on page 11 shows the **Specification** and **Results** tabs of the function block parameters GUI for a NCO.

Figure 13. Specification and Results Tabs of the Function Block Parameters GUI and the Results for a NCO



Real Mixer

The Real Mixer block performs a real-by-complex multiply on streams of data. It supports element-by-element multiplication on n channels and m frequencies.

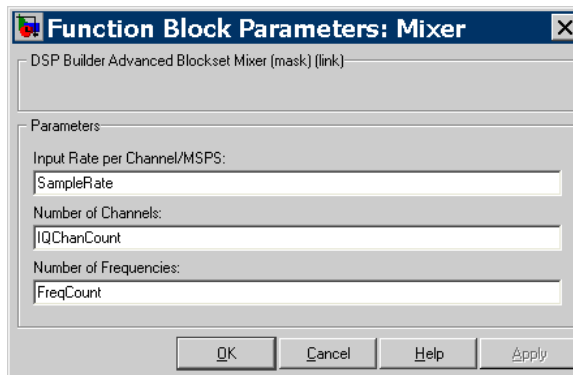
The Real Mixer equations are:

$$I_{out} = R_{in} * \cos(nco_phase);$$

$$Q_{out} = R_{in} * \sin(nco_phase);$$

Figure 14 shows the function block parameters GUI for a Real Mixer.

Figure 14. Function Block Parameters GUI for a Real Mixer.



Complex Mixer

The Complex Mixer block performs a complex-by-complex multiply on streams of data. It supports element-by-element multiplication on n channels and m frequencies.

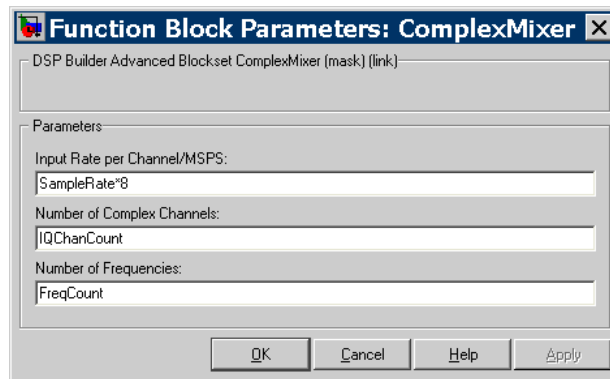
The Complex Mixer equations are:

$$I_{out} = I_{in} \cdot \cos(nco_phase);$$

$$Q_{out} = Q_{in} \cdot \sin(nco_phase);$$

Figure 15 shows the function block parameters GUI for Complex Mixer.

Figure 15. Function Block Parameters GUI for a Complex Mixer



For more details of the ModelIP blocks, refer to *DSP Builder Advanced Blockset User Guide* and *DSP Builder Advanced Blockset Reference Manual*.

Reference Designs

Reference designs that demonstrate the implementation of digital IF systems for wideband CDMA (W-CDMA) and WiMAX are provided with the DSP Builder Advanced Blockset. The reference designs for W-CDMA and WiMAX are located at `<install directory>\quartus\dspba\Examples\ReferenceDesigns\`

You must have matching versions (for example 8.0 or 8.0 SP1) of the Quartus® II software and DSP Builder. You must also have a compatible version of The MathWorks software that includes MATLAB, Simulink, and the Simulink Signal Processing blockset.

WiMAX Designs

This section outlines the system level design and hardware implementation of basestation configurations for WiMAX.

The sample rate per channel is specified in the WiMAX standard as 11.2 MHz. The clock frequency is 179.2 MHz, 16 times the sample rate.

The details of the filter design are outside the scope of this document. The filter chain for the DUC design is partitioned with a single rate channel filter, interpolate by two, and interpolate by four FIR filters.

Figure 16 shows the implementation of a DUC design with one I/Q complex channel.

Figure 16. DUC Design With One I/Q Complex Channel

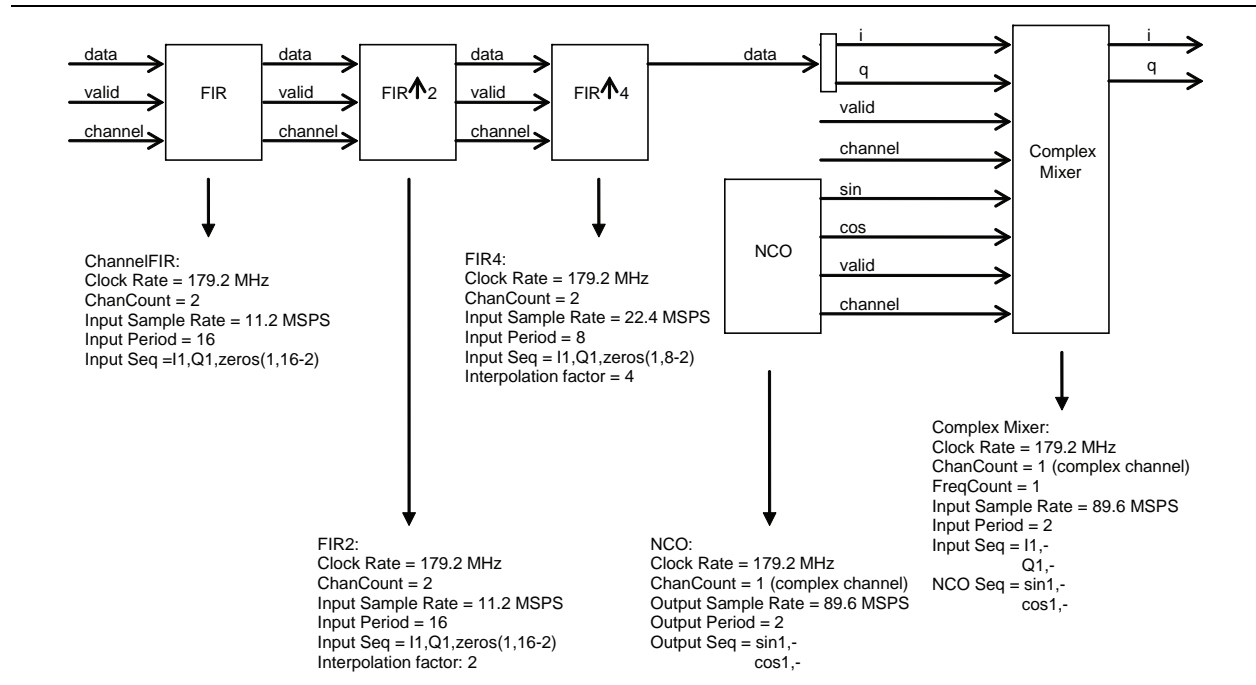
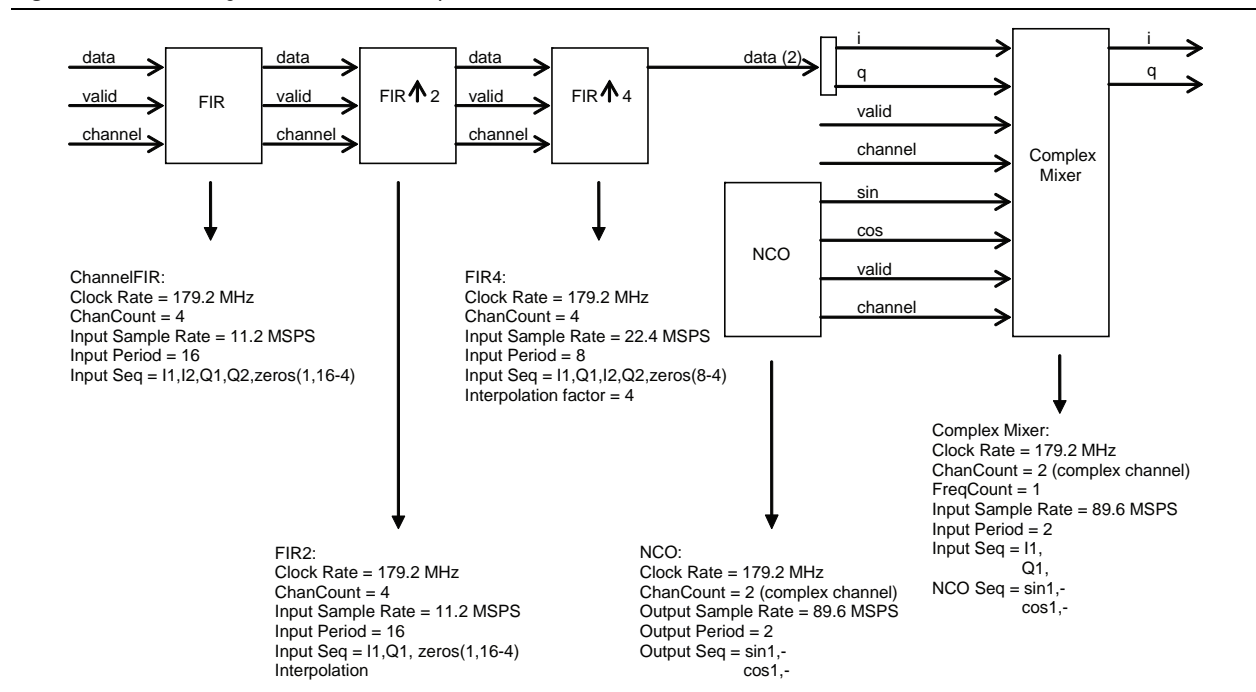


Figure 17 shows the implementation of a DUC design with two I/Q complex channels.

Figure 17. DUC Design With Two I/Q Complex Channels



The implementations of one and two I/Q complex channel DUC designs are very similar except for the connections between the filter chain output and the complex mixer. These connections were described in “[Multichannel TDM Operation](#)” on page 5.

The annotations under each block are useful for understanding the design flow.

For a high speed device family, you can easily double the clock rate and the tool meets the required timing through automatic pipelining. You can also increase the number of channels using a workspace variable without having to re-architect the whole system.



For more details of the WiMAX system requirements, refer to [AN421: Accelerating DUC & DDC System Designs for WiMAX](#).

W-CDMA Designs

This section outlines the system level design and hardware implementation of popular base station configurations for W-CDMA systems.

A popular W-CDMA base station configuration requires DDC and DUC associated with two diversity antennas, four carriers and three sectors. This is often referred to as a multichannel configuration.

An additional popular W-CDMA system supports low-capacity base stations that serve small indoor areas. This type of base station is often referred to as a picocell base station. A typical configuration consists of diversity antennas and a single carrier frequency.

The baseband input sample rate per channel for a W-CDMA system is 3.84 MHz. In the multichannel configuration, the FPGA clock frequency is chosen to be 245.76 MHz to support four carriers and two MIMO antennas (16 in-phase and quadrature channels in total). The details of the filter design are described in AN442 and are beyond the scope of this application note.

[Figure 18 on page 15](#) shows the implementation of the multichannel DUC design.

In the picocell design ([Figure 19 on page 15](#)) the FPGA clock is set to 122.88 MHz, and four in-phase and quadrature channels are supported.



For more information including the system specification for the W-CDMA IF modem designs, please refer to [AN442: Tool Flow for Design of Digital IF for Wireless Systems](#).

Figure 18. Multichannel W-CDMA DUC Design

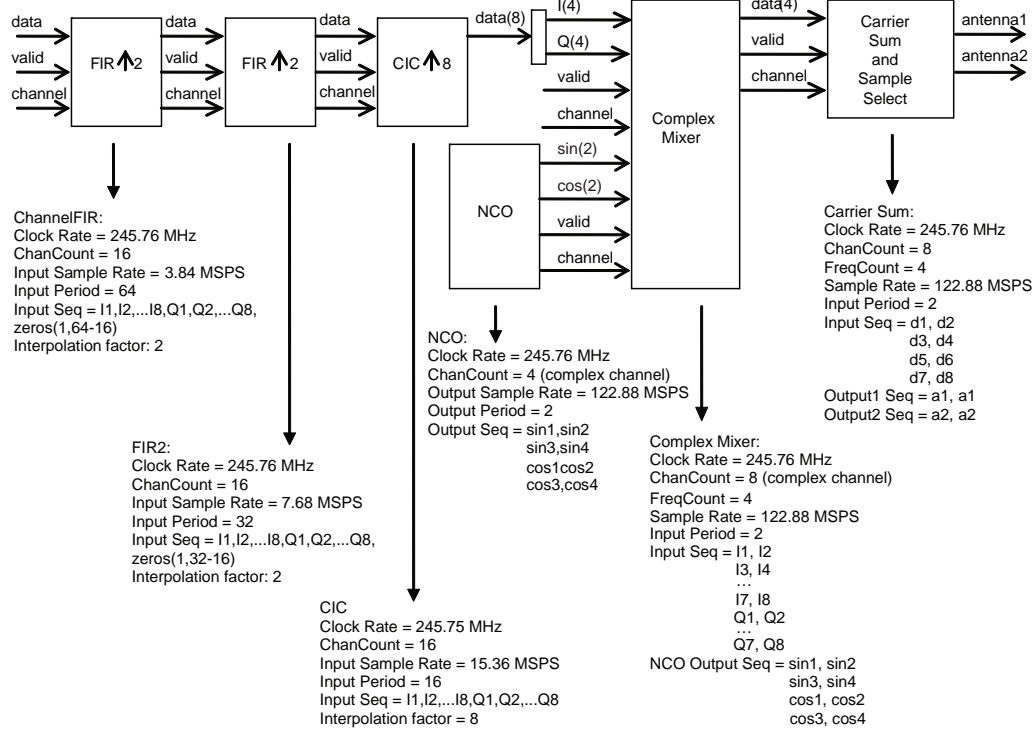
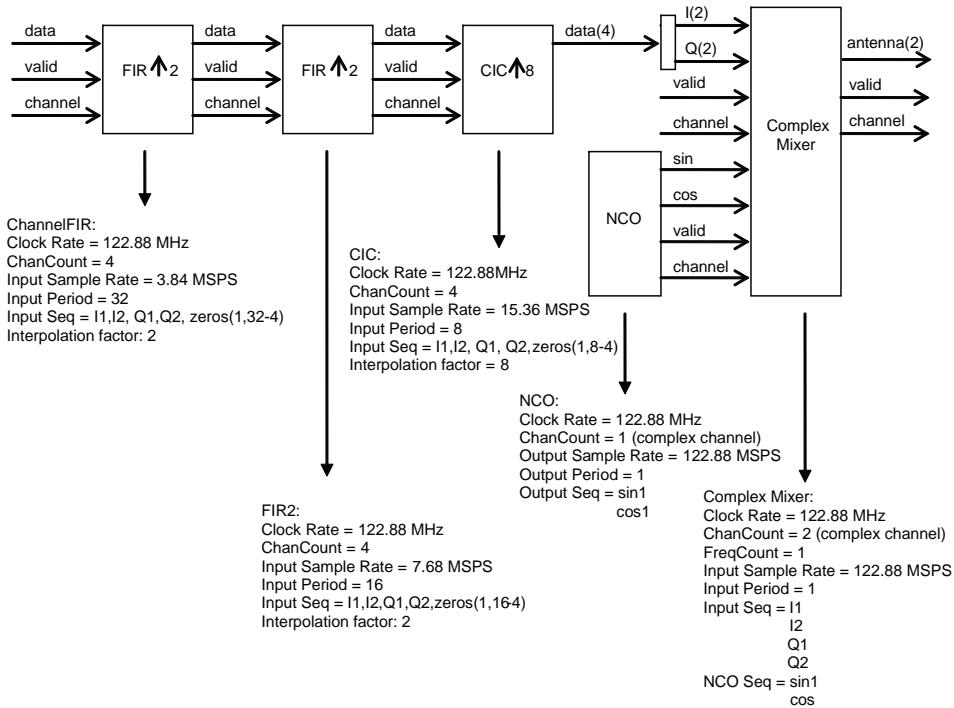


Figure 19 shows the implementation of the picocell DUC design.

Figure 19. Picocell W-CDMA DUC Design



Conclusions

The DSP Builder Advanced Blockset tool flow offers the following advantages:

- Easy to design flow: You can easily build a hardware design that has no FPGA specific elements and enter the design naturally. The unified interfaces make block connections easy. MATLAB/Simulink remains analysis tool of choice.
- Design exploration through system level constraints: You can configure the blocks in the data path with high level constraints. Then you just select the device target and compile the design.
- Increased performance and productivity: The designs are independent of library technology and automatically map to the target FPGA device making the designs portable and reusable. The tool also provides push-button optimized RTL and the automatic pipelining and folding enables timing closure.

Revision History

Table 1 shows the revision history for this application note.

Table 1. Template Revision History

Date	Changes Made
August 2008	Initial Release.



101 Innovation Drive
 San Jose, CA 95134
www.altera.com
 Technical Support
www.altera.com/support

Copyright © 2008 Altera Corporation. All rights reserved. Altera, The Programmable Solutions Company, the stylized Altera logo, specific device designations, and all other words and logos that are identified as trademarks and/or service marks are, unless noted otherwise, the trademarks and service marks of Altera Corporation in the U.S. and other countries. All other product or service names are the property of their respective holders. Altera products are protected under numerous U.S. and foreign patents and pending applications, maskwork rights, and copyrights. Altera warrants performance of its semiconductor products to current specifications in accordance with Altera's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Altera assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Altera Corporation. Altera customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.

