Multi-Channel Digital Up/Down Converter for WiMAX Systems

Introduction

Digital Up Converters (DUC) and Digital Down Converters (DDC) are widely used in communication systems for scaling the sample rates of signals. Digital up conversion is required when a signal is translated from baseband to intermediate frequency (IF) band. Digital down conversion happens when a signal is converted from intermediate frequency band to baseband. DUCs and DDCs typically include frequency shifting using mixers in addition to sampling rate conversion. The structure of a DUC or DDC depends on the conversion ratio required. For WiMAX (Worldwide Interoperability for Microwave Access) systems, the conversion ratio is typically 8 to 10. For such low conversion rates, the DUC or the DDC is constructed using FIR filters only. If the required conversion ratio is much higher, a Cascaded Integrator-Comb (CIC) filter is used in the DDC/DUC structure.

The Lattice Multi-Channel (MC) DUC/DDC reference design addresses the 10 MHz bandwidth channels for Wireless-MAN or Wireless-HUMAN PHY for WiMAX systems. An IF sampling frequency of 89.6 MHz is used for this reference design. The MC DDC/DUC is designed to support four complex channels (i.e., each channel having an I and a Q component). The MC DUC and MC DDC reference design packages include pre-configured Lattice ispLeverCore™ Intellectual Property (IP) Cores for the FIR (Finite Impulse Response) filters, DA-FIR (Distributed Arithmetic Finite Impulse Response) filters and the NCO (Numerically Controlled Oscillator).

Features

- Compliant with the WiMAX transmit spectral mask
- Four complex (I and Q channels) DDC/DUC for a sampling frequency of 89.6 MHz
- 18-bit data and coefficient widths with pre-determined coefficient values
- Includes evaluation test bench and scripts for simulation and implementation
- Includes scripts to create IPs and simulation libraries
- Self-checking test bench with programs to generate golden output
- Verilog source code is provided to enable customization of the design

WiMAX Requirements

The DUC/DDC reference design is designed to be used in WiMAX applications. The transmit spectral mask for WiMAX is defined in IEEE 802.16-2004 standard and reproduced in Figure 1. The corner frequencies for the 10 and 20 MHz bands are shown in Table 1. The interpolation and decimation filters for the DUC and DDC respectively are designed to satisfy the WiMAX spectral mask shown in Figure 1.

Figure 1. Transmit Spectral Mask for WiMAX

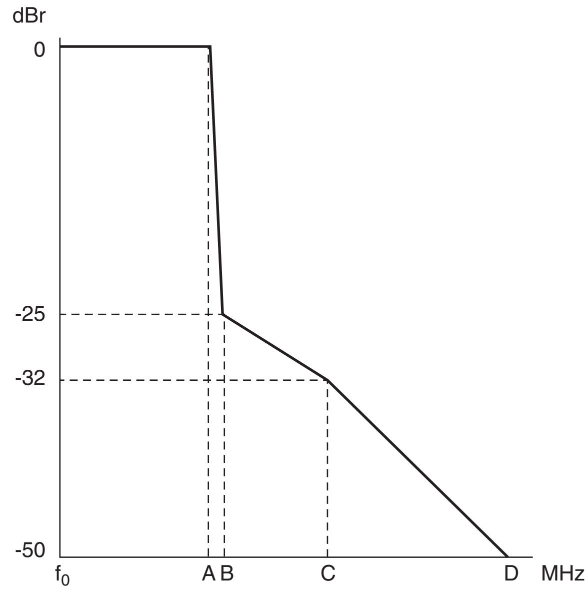


Table 1. Transmit Spectral Mask Parameters

Channelization (MHz)	A	B	C	D
20	9.5	10.9	19.5	29.5
10	4.75	5.45	9.75	14.75

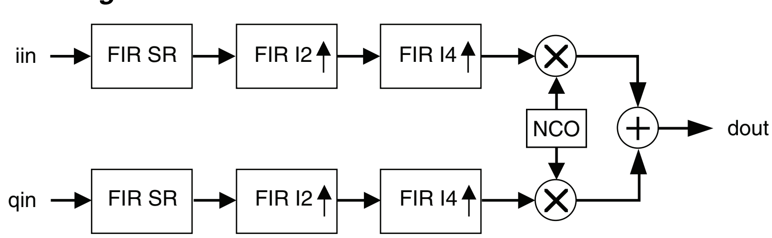
Functional Description

This section first describes the DUC in detail followed by the DDC. Since there are many similarities between the DUC and the DDC, the common theory is covered in the DUC section and omitted in the DDC section.

Digital Up Converter

Digital up converter (DUC) translates the base band signal to a higher frequency band. This is done by first up-sampling the base band signal to the required sampling frequency and then mixing it with a high-frequency carrier. A functional block diagram of the DUC is given in Figure 2.

Figure 2. Functional Block Diagram of the DUC

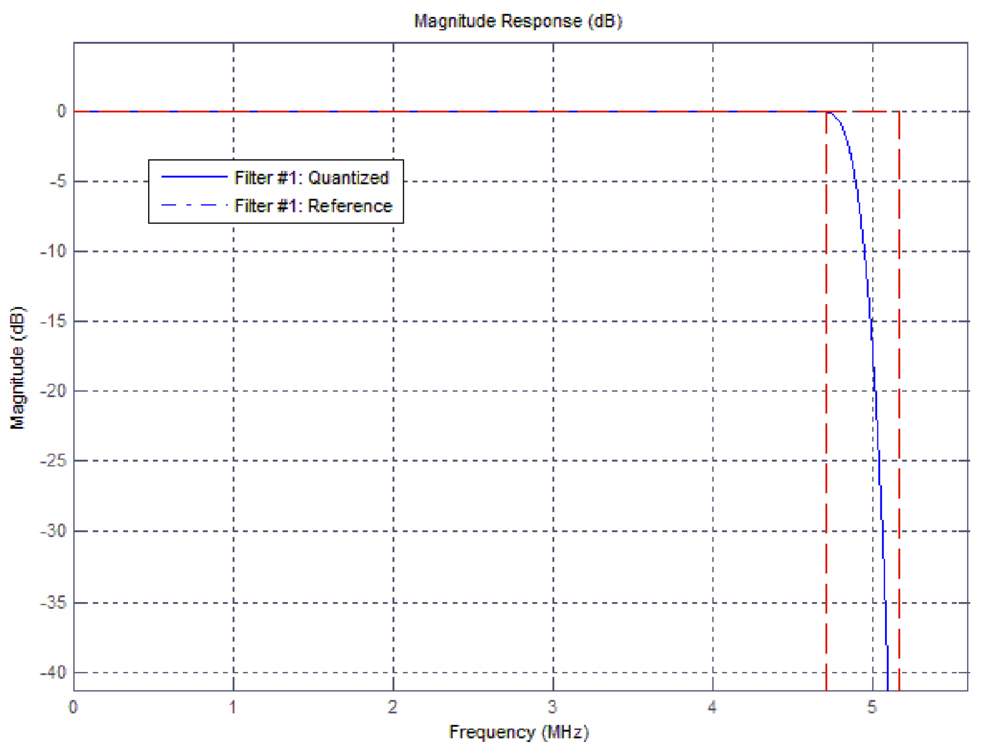


The DUC has two identical paths, one for the I-input and the other for the Q-input. For this reason, it is also referred to as a complex DUC. The base band signal is first up-sampled to the IF frequency of 89.6 Msps before mixing with an NCO output to produce the spectrum centered around the desired modulation frequency. The sampling rate of the input signal is assumed to be 11.2 Msps. This signal has to be up-sampled by a factor of 8 to achieve a sampling rate of 89.6 Msps. This up-sampling can be done in many different ways. For example, there can be a single interpolation filter that performs up-sampling by 8 or there can be three interpolation filters connected in cascade, each performing an up-sampling by a factor of 2. For a FPGA implementation, smaller resource utilization is achieved by appropriately factoring the interpolation filters to more than one stage.

The WiMAX standard requires a very sharp transition band between 4.75 MHz and 5.45 MHz with an attenuation of at least 25 dB. The stop band attenuation for practical implementations is usually required to be much higher, in the order of 90 dB, for successful operation in interference prone situations. The interpolation filter therefore needs to have a sharp transition characteristic and high attenuation in the stop band. Such a filter will become expensive due its high number of taps and the high sampling rate in which it operates. By splitting the interpolation filter into multiple stages, it is possible to achieve the same overall response using multiple filters with one having sharp transition characteristics and the others having progressively wider transition characteristics. The sharp transition realized using a filter with a large number of taps can be positioned in the left end (low sampling rate end) of the filter chain to reduce the effective FPGA resources used up by the filter. The filters at the right end of the chain operate at high sampling rates, but they have a fewer number of taps due to their relatively wide transition band. The optimal partitioning of the up-sampling filter chain is beyond the scope of this document. In this design, the interpolation filter is realized using a three-stage FIR filter chain consisting of a single-rate, sharp-transition FIR filter, an interpolation by 2 FIR filter and an interpolation by 4 FIR filter. In Figure 2, the notations SR, I2 and I4 are used to represent single rate, interpolation by 2 and interpolation by 4 filters, respectively.

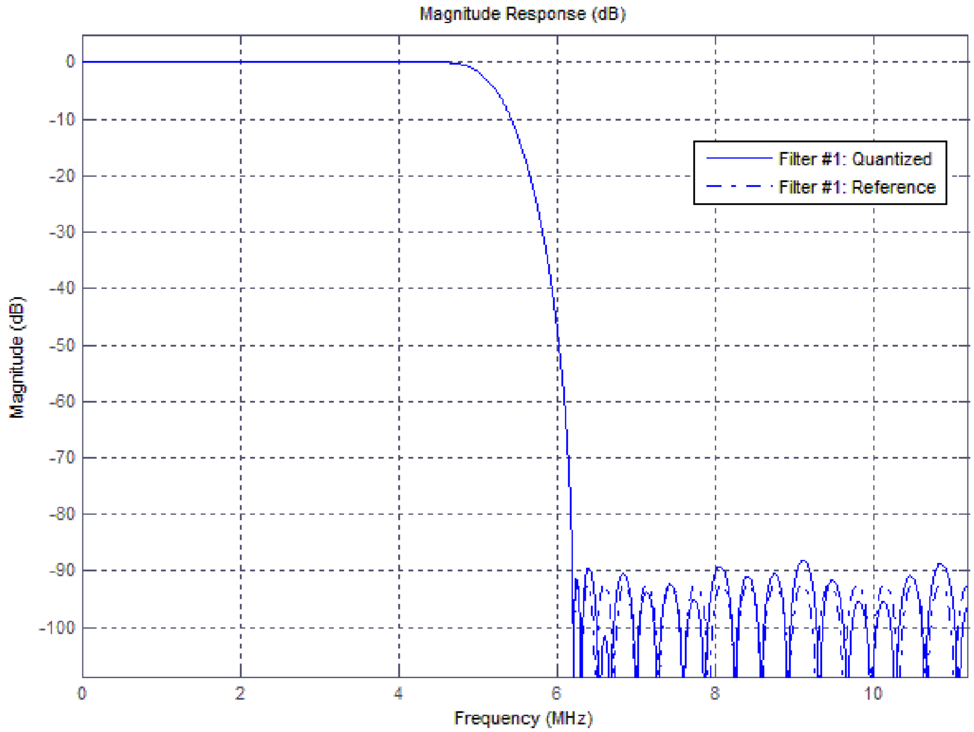
The first filter in the chain, FIR-SR, is a single-rate FIR filter that helps to achieve a sharp transition band. This filter is a 111-tap symmetrical, single-rate FIR filter. The frequency response of this single rate filter is shown in Figure 3. The filters were designed using the MATLAB® Filter Design Toolbox™. The results shown are for a 17-bit data and 18-bit coefficients quantization settings.

Figure 3. Frequency Response for the Single-Rate FIR Filter



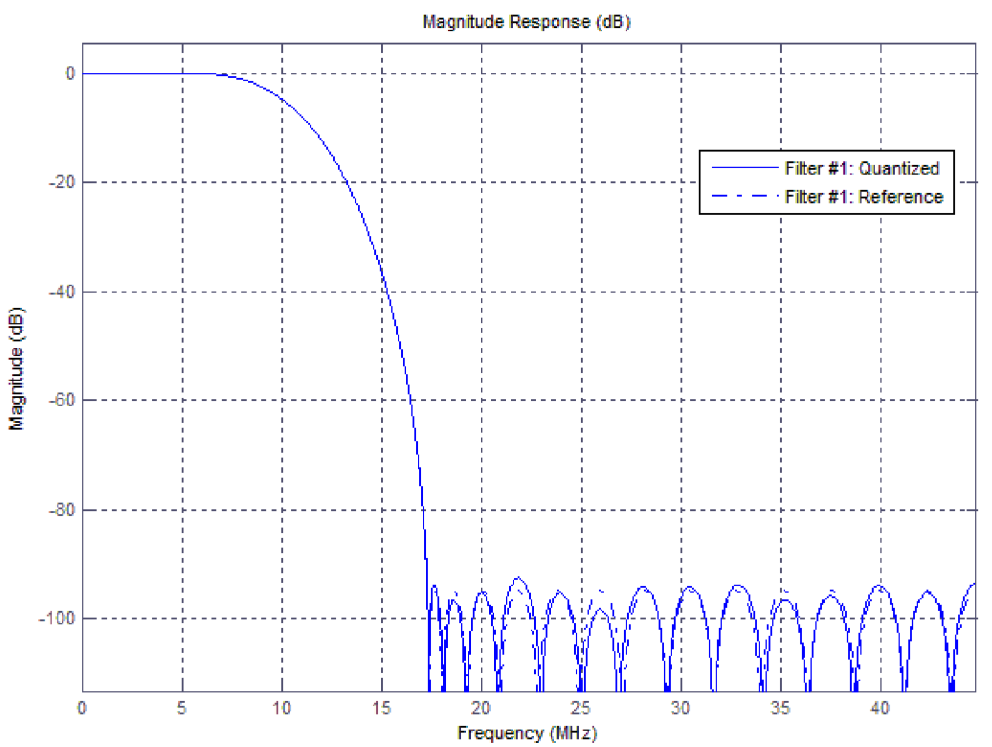
The second stage is an interpolating FIR with an interpolation factor of 2. This has a moderately sharp transition response and requires 63 taps. The frequency response for the interpolation by 2 FIR filter (FIR I2) is given in Figure 4. For this filter, the data and coefficients width are equal to 18 bits.

Figure 4. Frequency Response for the Interpolation by 2 FIR Filter



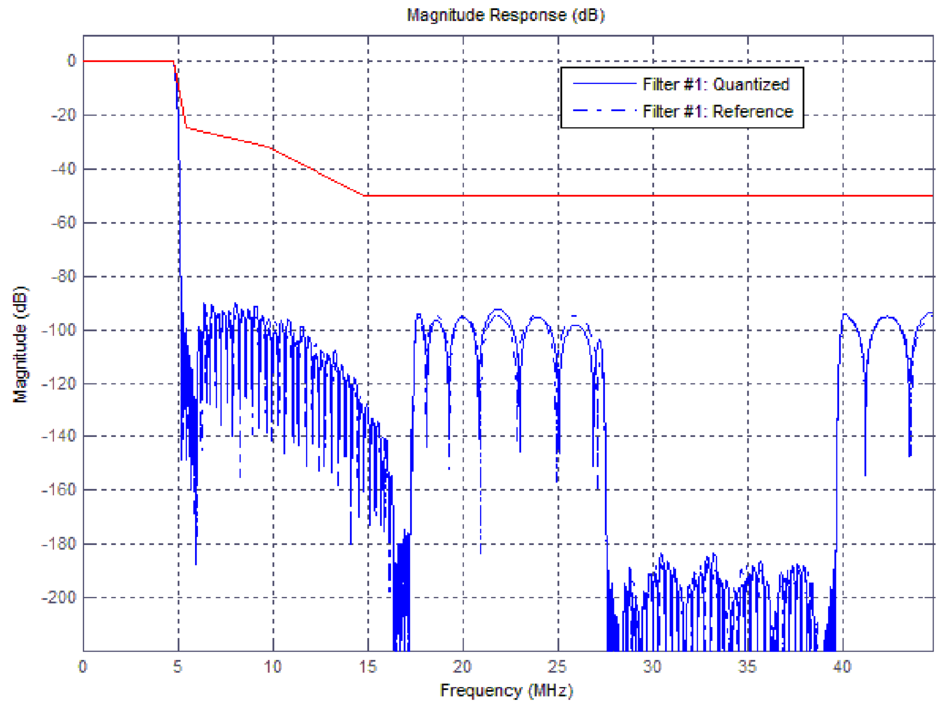
The third stage is a factor of 4 interpolation FIR filter (FIR I4) whose transition band is very wide and is realized using 35 taps. The frequency response for this filter is shown in Figure 5 for 18-bit data and coefficients widths.

Figure 5. Frequency Response for the Interpolation by 4 FIR Filter



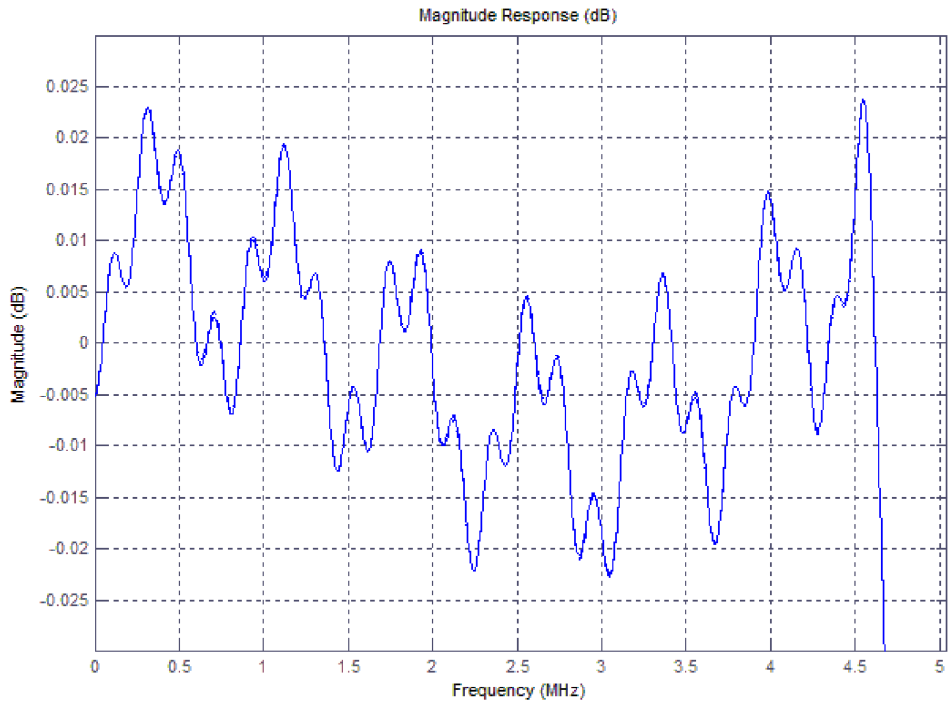
The three filters described above are cascaded to obtain the overall frequency response that satisfies the WiMAX transmit spectral mask requirements. Figure 6 shows the overall magnitude response with the WiMAX transmit spectral mask overlaid on it.

Figure 6. Magnitude Response for the DUC FIR Filter Chain



The WiMAX standard requires that the pass band ripple for the FIR filter chain be less than 0.05 dB. As seen in Figure 7, the pass-band ripple for this design is within the required range.

Figure 7. Pass-band Ripple for the DUC Filter Chain

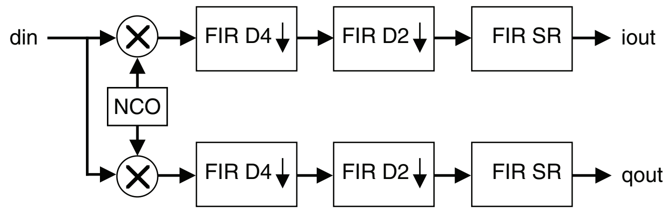


The up-sampled signal out of the last stage of the FIR filter is then mixed with the orthogonal sinusoidal outputs from an NCO: the sine and cosine components are multiplied with I and Q channels respectively. The outputs of the multipliers are added up to get the final DUC output.

Digital Down Converter

The digital down converter (DDC) performs the reverse function of that of a DUC. It converts a signal from the IF band to the base band. The DDC is built using a structure similar to the DUC, but it uses decimation filters instead of interpolation filters and they are connected in the reverse order of the DUC. A functional block diagram of the DDC is shown in Figure 8.

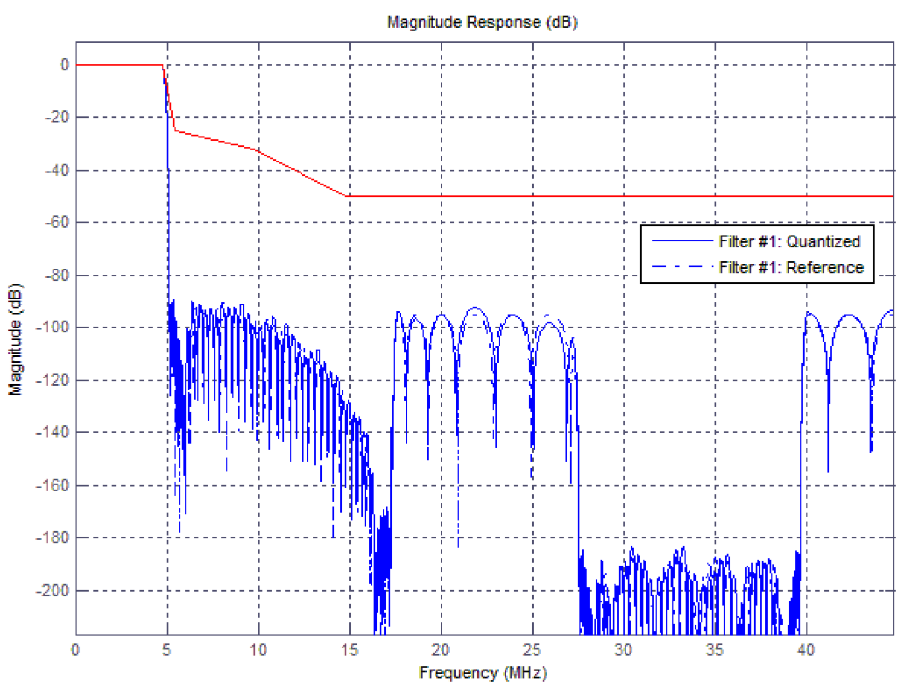
Figure 8. Functional Block Diagram of the DDC



As shown in Figure 8, the input signal is first mixed with the IF frequency sine and cosine outputs from the NCO. This creates multiply replicated base band frequency response spectrum. The filter chain that follows the mixer reduces the sampling rate and performs low pass filtering to remove the spectral replications and to yield the correct base band spectrum. In Figure 8, the notations D4, D2 and SR are used to denote decimation by 4, decimation by 2 and single rate filters, respectively. The first filter in the chain is a decimation by 4 FIR filter (FIR D4) which reduces the sampling rate by 4 and performs low pass filtering. The pass band to stop band transition is not sharp for this filter. The second stage is a decimation by 2 FIR filter (FIR D2) with a moderate transition band. The final filter in the chain is a single rate low pass filter whose main function is to provide a sharp transition band, so the stipulation in the WiMAX transmit mask is met.

The frequency responses of the FIR filters are very similar to those in the DUC filter chain. The overall magnitude response of the DDC filter chain along with the WiMAX spectral mask is given in Figure 9.

Figure 9. Magnitude Response for the DDC FIR Filter Chain



Implementation Description

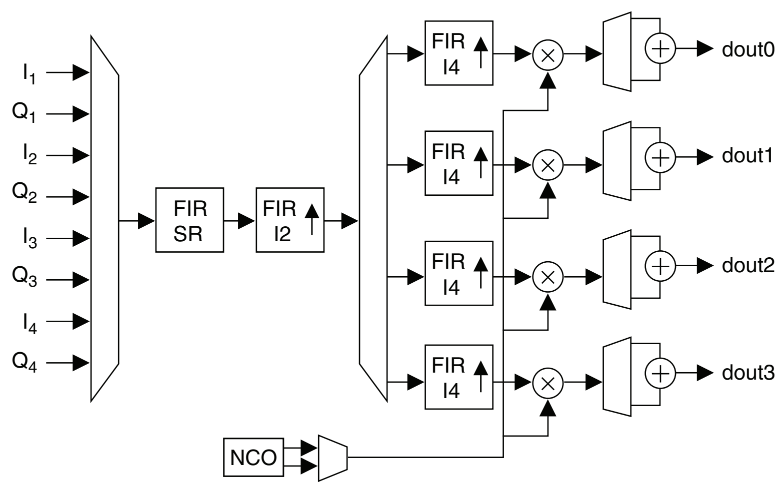
The MC DUC and MC DDC are implemented using a scheme that results in the best area utilization on FPGA devices. Since the filter chains in both I and Q channels are identical, the same set of filters can be time-multiplexed to accommodate both the channels. In this reference design, the sampling frequency is chosen to be 89.6 Msps and hence the time sharing of the filter chain between two channels requires the filters to operate at 179.2 Msps.

The MC DUC and MC DDC reference designs are built using the Lattice NCO version 2.3, FIR Filter version 4.0 and DA-FIR Filter version 2.0 IP cores.

Multi-Channel Digital Up Converter

The implementation diagram for the MC DUC is shown in Figure 10.

Figure 10. Implementation Diagram for the MC DUC



The FIR filter chain consists of the following three FIR filters connected in cascade:

1. Single-rate FIR filter (FIR-SR)
2. Interpolation by 2 FIR filter (FIR-I2)
3. Interpolation by 4 FIR filter (FIR-I4)

The FIR filters in the filter chain can be chosen to be either Mult FIR or DA-FIR or a combination of the two. The choice depends on the availability of DSP blocks and LUTs in the selected device. In this design, we have chosen Mult-FIR to implement FIR SR and DA-FIR to implement FIR I4 and I2, so that the design could be implemented on a LatticeECP2M-35 device.

The parameters for these filters are given in Table 2.

Table 2. IP Parameters for the FIR Filters in the MC DUC Filter Chain

Parameter	FIR-SR	FIR-I2	FIR-I4
IP type	FIR filter	DA-FIR filter	DA-FIR filter
Number of channels	8	8	2
Number of taps	111	63	35
Filter type	Single rate	Interpolator	Interpolator
Interpolation factor	—	2	4
Symmetric coefficients	Yes	Yes	Yes
Number of multipliers	28	—	—
Input data type	Signed	Signed	Signed
Input width	17	18	18
Input binary point	15	16	16
Coefficient data type	Signed	Signed	Signed
Coefficient width	18	18	18
Coefficient binary point	17	17	17
Output width	18	18	18
Output binary points	14	15	15
Overflow	Saturation	Saturation	Saturation
Rounding	Convergent	Convergent	Convergent

As shown in Figure 10, the inputs iin and qin are multiplexed and applied at the input of FIR-SR consistent with the ibstart input of the FIR-SR IP. The output from FIR-SR is demultiplexed and routed to separate registers for each channel. This data is then read out and multiplexed consistent with the ibstart input from the second stage interpolation FIR filter (FIR-I2) and applied at the inputs of FIR-I2. The interpolated output from FIR-I2 is demultiplexed and applied to the input of the next stage interpolation-by-4 FIR filter (FIR-I4). The outputs of FIR-I4 are directly connected to the mixer.

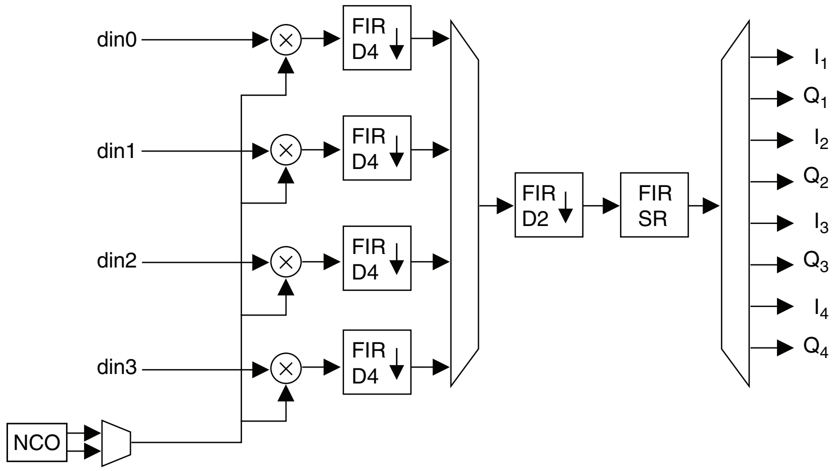
Due to the time-multiplexed nature of the I and Q data from the FIR filter chain, the mixer needs to multiply the filter chain output with the sine and cosine outputs during alternate cycles. The NCO is configured to have sine and cosine outputs, but made to operate at half the clock speed, i.e., 89.6 MHz. The sine and cosine outputs are multiplexed and applied to the multiplier inputs. Every two consecutive outputs of the multiplier are added together by using a register and using a clock enable on the adder.

The output from each of the filters in the filter chain becomes valid after a latency period. The final output from the MC DUC therefore has a latency which is the sum of the latencies of the individual IPs. The output latency of the MC DUC reference design is around 380 clock cycles.

Multi-Channel Digital Down Converter

The implementation diagram for the MC DDC is shown in Figure 11.

Figure 11. Implementation Diagram for the MC DDC



The FIR filter chain for the MC DDC consists of the following three FIR filters connected in cascade:

1. Decimation by 4 FIR filter (FIR-D4)
2. Decimation by 2 FIR filter (FIR-D2)
3. Single-rate FIR filter (FIR-SR)

Similar to the DUC implementation, it is possible to use any combination of Mult-FIR and DA-FIR for the FIR filters in the filter chain. In this design, we have chosen Mult-FIR to implement FIR SR and DA-FIR to implement FIR D4 and D2.

The parameters for these filters are given in Table 3.

Table 3. IP Parameters for the FIR Filters in the MC DDC Filter Chain

Parameter	FIR-D4	FIR-D2	FIR-SR
IP type	FIR filter	DA-FIR filter	DA-FIR filter
Number of channels	2	8	8
Number of taps	35	63	111
Filter type	Decimator	Decimator	Single rate
Interpolation factor	4	2	-
Symmetric coefficients	Yes	Yes	Yes
Number of multipliers	-	-	28
Input data type	Signed	Signed	Signed
Input width	18	18	17
Input binary point	16	16	15
Coefficient data type	Signed	Signed	Signed
Coefficient Width	18	18	18
Coefficient binary point	19	18	17
Output width	18	18	18
Output binary points	16	15	14
Overflow	Saturation	Saturation	Saturation
Rounding	Convergent	Convergent	Convergent

The implementation scheme for the MC DDC is very similar to that of the MC DUC. The NCO output multiplexing scheme is the same as the MC DUC. The output of the multiplier has alternating data for the channels. They are directly connected to the decimation-by-4 FIR (FIR-D4) filter. The output of FIR-D4 is multiplexed to obtain the inputs to the decimation-by-2 FIR (FIR-D2) filter. The output of FIR-D2 is directly connected to the input of the single-rate FIR (FIR-SR). The output of FIR-SR is demultiplexed and given out at iout and qout ports.

The output latency for the MC DDC reference design is about 560 cycles.

Signal Descriptions

The top-level interface diagrams for the MC DUC and the MC DDC are shown in Figures 12 and 13 respectively.

Figure 12. Interface Diagram for the MC DUC

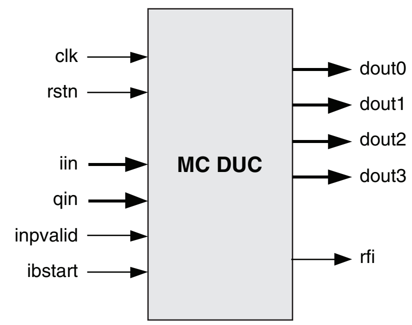
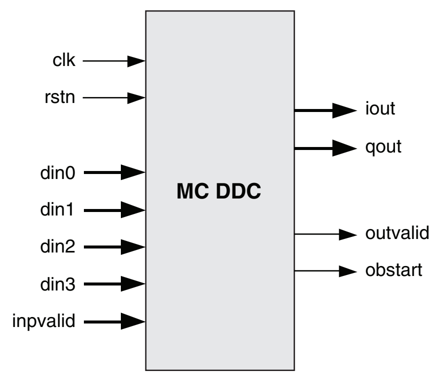


Figure 13. Interface Diagram for the MC DDC



A description of the Input/Output (I/O) ports for the MC DUC and the MC DDC reference designs are provided in Tables 4 and 5 respectively.

Table 4. Top-level I/O Interface for the MC DUC

Port	Bits	I/O	Description
clk	1	I	System clock. This must be two times the sampling rate. In this case, 2 x 89.6MHz =179.2 MHz.
rstn	1	I	System-wide active low reset. This input resets all the logic and IP cores in the MC DUC design.
inpvalid	1	I	Input valid. This signal must be asserted high to indicate a valid data on the iin and qin inputs.
ibstart	1	I	Input block start. This input identifies channel 0 of the input.
iin	17	I	I channel input data.
qin	17	I	Q channel input data.
rfi	1	O	Ready for Input. This signal must be high when a valid input data is given at iin and qin inputs. The rfi signal can be directly used to drive the inpvalid input.
dout0	19	O	Channel 0 output data. The output data rate is half the clock rate (i.e., a new output data comes out once every other clock cycle).
dout1	19	O	Channel 1 output data. The output data rate is half the clock rate (i.e., a new output data comes out once every other clock cycle).
dout2	19	O	Channel 2 output data. The output data rate is half the clock rate (i.e., a new output data comes out once every other clock cycle).
dout3	19	O	Channel 3 output data. The output data rate is half the clock rate (i.e., a new output data comes out once every other clock cycle).

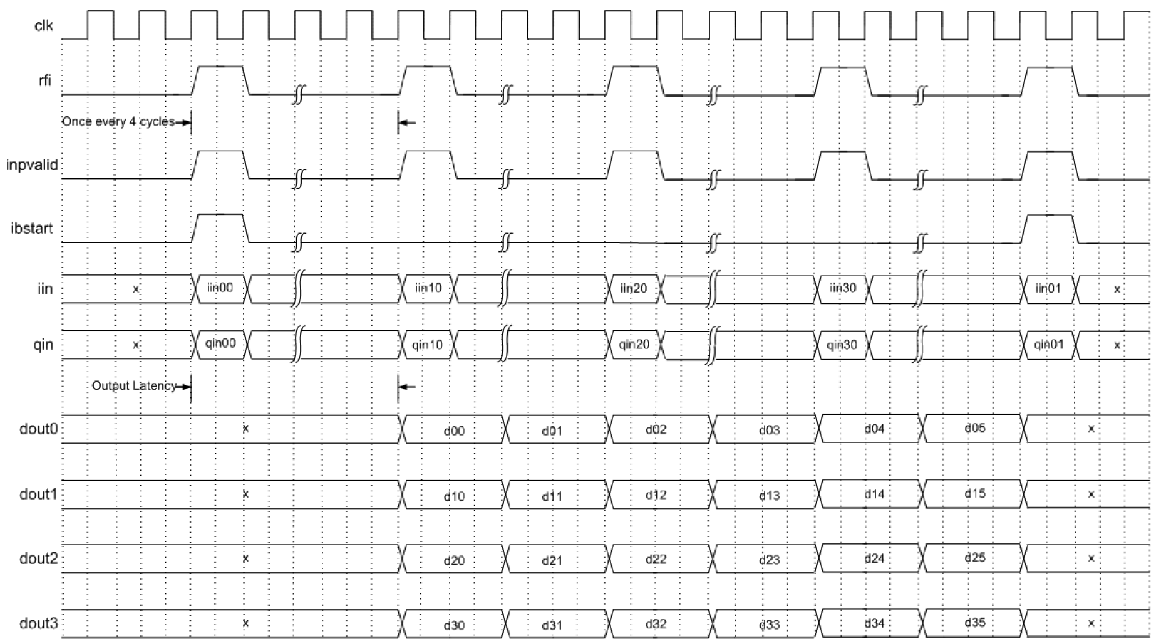
Table 5. Top-level I/O Interface for the MC DDC

Port	Bits	I/O	Description
clk	1	I	System clock. This must be two times the sampling rate. In this case, 2 x 89.6MHz = 179.2 MHz.
rstn	1	I	System-wide active low reset. This input resets all the logic and IP cores in the MC DDC reference design.
din0	18	I	Input data to channel 0 of the MC DDC.
din1	18	I	Input data to channel 1 of the MC DDC.
din2	18	I	Input data to channel 2 of the MC DDC.
din3	18	I	Input data to channel 3 of the MC DDC.
inpvalid	1	I	Input valid. This input is only used in simulation to indicate the first valid data, otherwise pulled up.
outvalid	1	O	Output valid. The output data at iout and qout are valid only when outvalid is high.
obstart	1	O	Output block start. This output identifies channel 0 of the output.
iout	18	O	I channel output data from the MC DDC. The output data rate is one-fourth the clock rate (i.e., a new output data comes out once every four clock cycles).
qout	18	O	Q channel output data from the MC DDC. The output data rate is one-fourth the clock rate (i.e., a new output data comes out once every four clock cycles).

Timing Diagrams

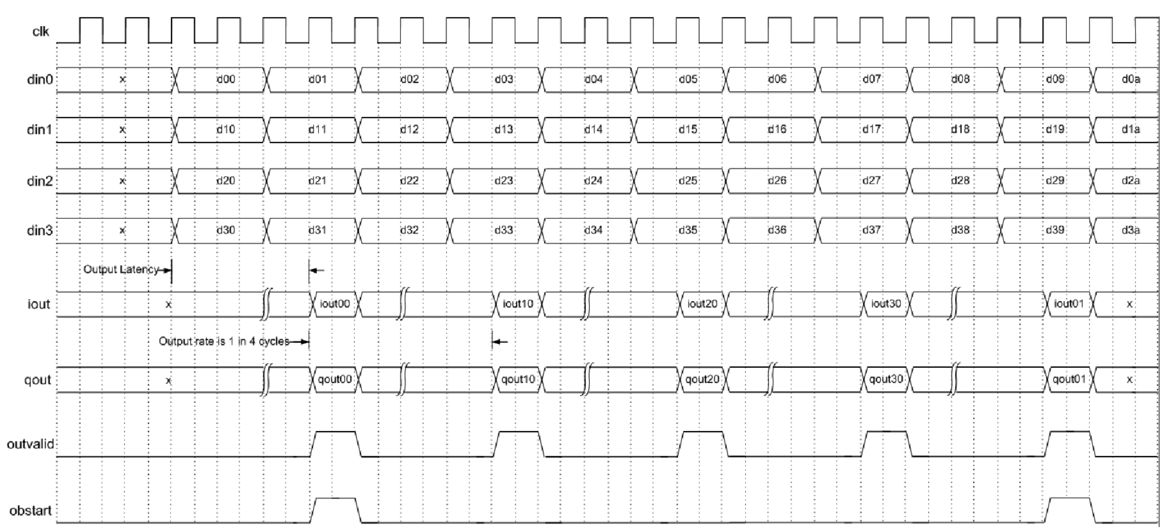
The timing diagrams for the MC DUC and MC DDC systems are shown in Figures 14 and 15 respectively.

Figure 14. Timing Diagram for the MC DUC[1]



1. In iinx$_1$x$_2$, qinx$_1$x$_2$ and dx$_1$x$_2$, x$_1$ stands for channel number and x$_2$ stands for data number.

Figure 15. Timing Diagram for the MC DDC[1]



1. In ioutx$_1$x$_2$, qoutx$_1$x$_2$ and dx$_1$x$_2$, x$_1$ stands for channel number and x$_2$ stands for data number.

Verifying and Implementing the Reference Design

The MC DUC/DDC reference design packages contain all the necessary design files and scripts to generate, simulate and implement the reference designs. The filter design itself was done using MATLAB and the parameter and coefficient files required for the reference design have been pre-generated and are available in the package.

The reference design distribution has independent packages for the MC DUC and the MC DDC designs. Both the designs follow the following directory structure.

```
<project_root>
|-----document
|-----impl
       |-----synplicity
|-----ips
       |-----du(d)cnco
       |-----dafiri(d)2
       |-----dafiri(d)4
       |-----firsr
|-----scripts
|-----sim
       |-----modelsim
       |-----aldec
|-----software
       |-----windows
              |-----bin
              |-----run
|-----src
|-----testbench
```

The document directory contains this user's guide and a readme text file with implementation information. The directory, 'impl' is where the Lattice ispLEVER® project and implementation files will reside. The directory "ips" contains all the IPs used in the reference design. For the MC DUC, the IPs NCO (under ducnco), FIR filter- single rate (under firsr), DA-FIR filter- Interpolate by 2 (under dafiri2) and DA-FIR filter- Interpolate by 4 (dafiri4) are available. For the MC DDC, the IPs NCO (under ddcnco), FIR filter- single rate (under firsr), DA-FIR filter-Decimate by 2 (under dafird2) and DA-FIR filter-Decimate by 4 (dafird4) are available. The directory, 'scripts' contains the scripts required for generating the IPs, creating simulation libraries for the IPs, creating test cases and generating implementation files. This folder also contains the top-level simulation script and the implementation project file. The 'sim' directory is setup for running the top-level simulation. The program files required to generate the expected output from the filter chain are available under the 'software' directory.

A master batch file called run_duc_mc.bat (or run_ddc_mc.bat) available under the project root directory serves as a push-button method to generate IPs, test cases and simulation files. The user needs to update the ispLEVER install directory path, ispLeverCore install directory path, ModelSim® executable path, Active-HDL® executable path, the current project directory path and the simulation tool before running this batch file. This batch file performs the following tasks:

- Creates the scripts required to generate the NCO, FIR and DA-FIR IP configurations using IPexpress™.
- Generates the NCO, FIR and DA-FIR IP configurations using IPexpress flow.
- Generates the ModelSim or Active-HDL simulation work libraries for running the top-level simulation.
- Runs the programs to generate the input stimulus and golden output data files
- Copies the required files to appropriate directories to run the top-level simulation and to run the FPGA implementations.

After running the run_duc_mc.bat (or run_ddc_mc.bat) batch file, all the required configurations are generated and the design is ready for top-level simulation and implementation. The master script may open multiple instances of ModelSim when it creates the work libraries for the IP cores. This may cause a license checkout error if there are not enough ModelSim licenses available. This error can be ignored and the ModelSim window can be closed, without affecting the work library creation process.

Running Functional Simulation

The reference design package provides the necessary environment for simulating the design in Mentor Graphics® ModelSim or Aldec Active-HDL. The evaluation test bench included in the package applies random data to the inputs and compares the output with the expected golden output. The golden output data is generated automatically by the scripts by running the model programs for each of the IPs in succession. As described previously, the master script, "run_duc_mc.bat" or "run_ddc_mc.bat, needs to be run first to generate the required simulation libraries. If the Modelsim (Aldec) simulation libraries (the populated work directory under <project_root>\ips\<ip_name>\sim\modelsim(aldec)) have not been generated, they need to be generated first before running the top-level simulation.

The functional simulation can be run using ModelSim SE by following the steps below:

1. Open Modelsim
2. Select the menu option **File -> Change Directory**
3. Set the directory to **<Project_root>/sim/modelsim**
4. Select the menu option **Tools -> TCL -> Execute Macro**
5. Select the file **ductop_sim.do** or **ddctop_sim.do**. The simulation process will then start and end with a pass or fail test result.

The functional simulation can be run using Aldec Active-HDL by following the steps below:

1. Open Aldec Active-HDL
2. Select the menu option **Tools -> Execute Macro**
3. Select the file **ductop_sim.do** or **ddctop_sim.do** under **<project_root>\ips\<ip_name>\sim\ aldec**. The simulation process will then start and end with a pass or fail test result.

The top-level simulation compiles the test bench, top-level MC DUC or MC DDC source file, a few other source files for the glue logic available under <project_root>\src directory and the IP black box module files from <project_root>\IPs\<ip_name> directories. Then the test bench is simulated with switches to include the simulation libraries for the four IPs. The test bench is self-checking and it will display the status of the tests at the end of the simulation run.

The implementation results for the MC DUC and MC DDC reference designs are given in Appendix A.

Implementing the Design on a LatticeECP2M™ Device

All the files necessary for the implementation must be created and moved to the right folders, when the master script "run_duc_mc.bat" or "run_ddc_mc.bat" is executed. If the script does not execute completely, some files may have to be copied manually to the <Project_root>/impl/synplicity directory. The implementation can be run by performing the following steps:

1. Run ispLEVER Project Navigator.
2. Select the menu option **File -> Open Project**.
3. Browse to **<project_root>\impl\synplicity** and select the project file **duc_top.syn** or **ddc_top.syn**.
4. When the project loads, all the necessary source files are added to the project and it is ready for the implementation process.
5. If necessary, change the logical preference file, duc_top.lpf or ddc_top.lpf as needed.

6. Run the implementation processes, Build Database, Map Design and Place & Route Design, in the usual way.

7. Check the map and timing reports to make sure they are acceptable.

Before running the implementation, the device size or speed grade may be changed if needed. Both the MC DUC and the MC DDC are required to operate with a 179.2 MHz clock. So, if the default options set in the project do not yield this frequency in the timing report, the preferences or other map, place and route options need to be changed to achieve this desired maximum frequency.

Customizing the Reference Design

The MC DUC/DDC reference design is complete with all the necessary files and scripts and it can be directly verified and implemented as explained in the previous sections. This section gives some information on making minor changes to the design to customize for your needs.

It may be required to make changes to the top-level RTL file, as well as some scripts to make it work for a new configuration. The amount of modification required depends on the type of changes that will be made to the design. If any of the IPs are changed, even for minor changes like NCO frequency or filter coefficients change, they have to be regenerated and the master script has to be run again to use the included push-button simulation or implementation flow. An easier way to do this is to create a new parameter file (<ip_name>.lpc) for the IP whose parameter has to be changed using the IP GUI in a separate directory. Then the existing parameter file can be replaced with the newly-generated file and the master script is run to generate the required IP files.

Changing the NCO Output Frequency

The current implementation uses a constant frequency output for the NCO IP. The frequency output of the NCO can be set as desired by changing the phase increment value in the NCO GUI. The phase increment may also be set to be variable. If the NCO settings are changed, it has to be regenerated before doing the top-level simulation or implementation. The generated NCO should be a single channel design with output width of 18 bits to work properly with the other modules in the package.

Changing the Parameters for the Filter Chains

This implementation uses one mixer frequency for four channels. The mixer frequency can be set differently for each channel by changing the NCO IP to work in Multi-Channel mode or adding multiple NCO IPs. If these settings are changed, the IP has to be regenerated and the top-level RTL file needs to be modified before doing the top-level simulation or implementation.

This package can be used without any major changes if only the filter coefficients have to be changed. If the number of channels, number of taps, number of multipliers, symmetry option, interpolation/decimation factor and the input/output widths are the same, no change will be needed to the top-level or script files. If the data widths are changed, top-level module and some of the glue logic instantiated inside the top-level need to be changed corresponding to the changed width. If only the coefficients are different, the existing coefficients file can be replaced by the new coefficients file and the FIR or the DA-FIR IP has to be regenerated for the new coefficients file.

Cautions on Customization

Making even minor modifications to the design or the constituent IPs can change the behavior of the Multi-Channel DUC/DDC design. It is the user's responsibility to make sure the changes made will not affect the functionality of the design. The user needs to thoroughly test the system as the functionality may easily become incorrect. If proper care is not taken, these problems may not be detected by the simulation or implementation flow provided in the package.

References

- IEEE Std. 802.16- 2004. IEEE Standard for Local and Metropolitan Area Networks - Part 16: Air Interface for Fixed Broadband Wireless Access Systems.

Technical Support Assistance

Hotline: 1-800-LATTICE (North America)
+1-503-268-8001 (Outside North America)
e-mail: techsupport@latticesemi.com
Internet: www.latticesemi.com

Revision History

Date	Version	Change Summary
April 2009	01.0	Initial release.

Appendix A. Implementation Results for LatticeECP2M Devices

Table 6. Performance and Resource Utilization[1]

Configuration	Slices	LUTs	Registers	18x18 MULTs	sysMEM™ EBRs	f_{MAX} (MHz)
MC DUC	8596	14966	16655	32	91	206.5
MC DDC	9750	18040	17964	32	91	206.9

1. Performance and utilization characteristics are generated targeting LFE2M-50E-7F484C using Lattice ispLEVER 7.2 software. When using this IP core in a different density, speed, or grade within the LatticeECP2M family or in a different software version, performance may vary.