Uplink Desubchannelization for WiMAX

February 2007, version 1.0 Application Note 450

Introduction

Altera provides building blocks to accelerate the development of a worldwide interoperability for microwave access (WiMAX) compliant basestations. This application note describes a reference design that demonstrates the suitability of the Altera® tools and devices for implementing the uplink desubchannelization function.

WiMAX is an emerging broadband wireless technology that promises high-speed data services. The *IEEE 802.16e-2005* standard enables mobility. There is significant market potential for this technology and it is currently being deployed by equipment manufacturers. Altera devices are the ideal platform for high throughput DSP designs such as those found on a WiMAX basestation channel card, because of the dedicated multiplier blocks and inherent parallel structure. This structure gives a significant cost and performance advantage over general purpose processors for this type of design.

For more information on *IEEE 802.16e-2005*, refer to the *IEEE Standard for Local and Metropolitan Area Networks, Part 16: Air Interface for Fixed Broadband Wireless Access Systems, IEEE P802.16e-2005, February 2006.*

In orthogonal frequency-division multiple access (OFDMA) systems, multiple end stations or subscriber stations (SSs) transmit at the same time to the access point (AP) or basestation. The uplink bandwidth is split into different subchannels, where each subchannel occupies different frequencies in the available bandwidth. Each SS is allocated one or more subchannels to allocate their data on. Some subchannels are allocated for any SS to transmit on. These tend to be used for ranging (timing synchronization) between SS and AP. Hence on these subchannels there may be contention between different SSs.

At the AP, the received signal is downconverted, the cyclic prefix is removed and converted into the frequency domain using an FFT operation, and the signal is split up into its separate subchannels—desubchannelization.

The uplink desubchannelization reference design is for the AP on the uplink (receiver path). It accepts the frequency domain symbol data and extracts the different subchannels from it. It then outputs user data on one output interface and ranging data on a separate output interface.

The uplink desubchannelization reference design provides the following features:

- Desubchannelization functions compliant to mandatory parts of *IEEE802.16e-2005 specifications:*
  - Supports UL-PUSC mode
  - No support for mini-subchannels
  - Suitable for use in WiMAX compliant OFDMA basestations
- Slot data extracted from input OFDMA symbols is output on a dedicated output interface:
  - A slot is one subchannel over three consecutive OFDMA symbols
  - Both data and pilot information output
- Ranging data extracted from input OFDMA symbols is output on its own dedicated output interface
- Configuration interface:
  - Allows you to specify the number of ranging channels in the next three OFDMA symbols and to specify which subchannels are allocated to each ranging channel
  - Allows you to specify which subchannels are allocated for user data including contiguous and non-contiguous subchannel ranges
  - `IDCell` specified on input port
- Supports all FFT sizes (128, 512, 1,024, and 2,048) as a synthesis time parameter
- Support for multiple antennas

The uplink desubchannelization reference design is compliant with the following WiMAX specification versions:

- *IEEE P802.16-Revd/D5-2004 "Part 16: Air Interface for Fixed Broadband Wireless Access Systems"*
- *IEEE Std 802.16e-2005 & IEEE Std 802.16-2004/Cor 1-2005 "Part 16: Air Interface for Fixed and Mobile Broadband Wireless Access Systems Amendment 2: Physical and Medium Access Control Layers for Combined Fixed and Mobile Operation in Licensed Bands and Corrigendum 1"*

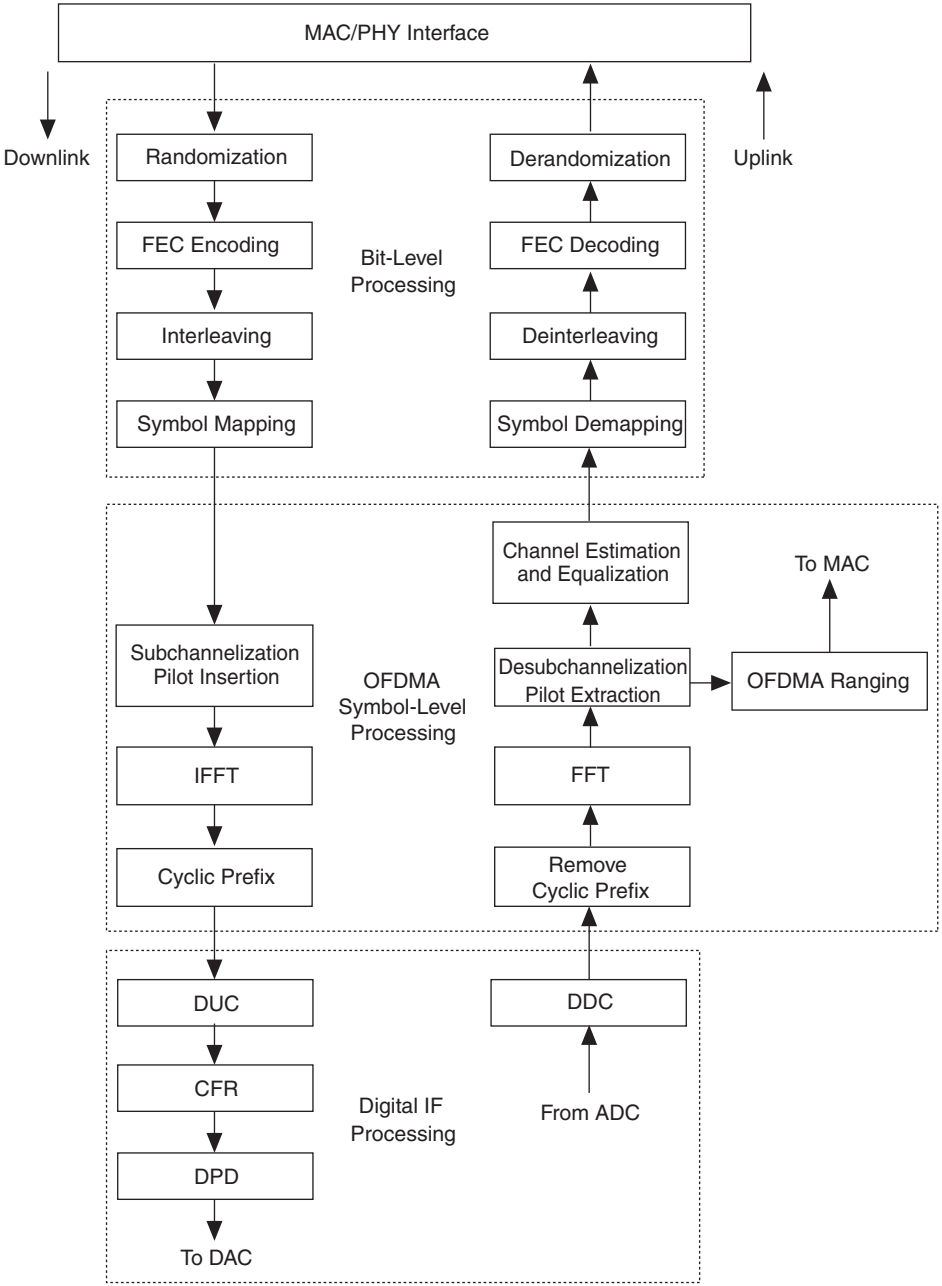
The design complies to the following sections of the two specifications:

- *8.4.6.2.1 Symbol structure for subchannel (PUSC)*
- *8.4.6.2.2 Partitioning of subcarriers into subchannels in the uplink*
- *8.4.6.2.3 Uplink permutation example*

WiMAX Physical Layer

Figure 1 shows an overview of the *IEEE 802.16e-2005* scalable OFDMA physical layer (PHY) for WiMAX basestations.

Figure 1. WiMAX PHY Implementation



Altera's WiMAX building blocks include bit level, OFDMA symbol-level, and digital intermediate frequency (IF) processing blocks. For bit-level processing, Altera provides symbol mapping reference designs and support for forward error correction (FEC) using the Reed-Solomon and Viterbi MegaCore® functions.

The OFDMA symbol-level processing blocks include reference designs that demonstrate subchannelization and desubchannelization with cyclic prefix insertion supported by the fast Fourier transform (FFT) and inverse FFT (IFFT) MegaCore functions. Other symbol-level reference designs illustrate ranging, channel estimation, and channel equalization.

The digital IF processing blocks include single antenna and multi-antenna digital up converter (DUC) and digital down converter (DDC) reference designs, and advanced crest factor reduction (CFR) and digital predistortion (DPD).

This application note describes uplink desubchannelization.

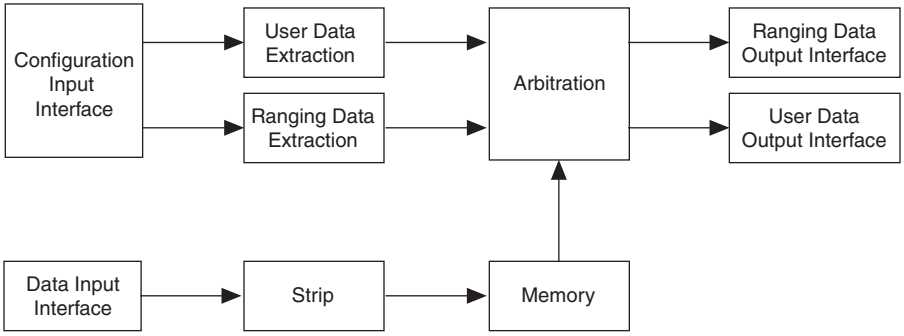
For more information on Altera WiMAX solutions, refer to the following application notes:

- *AN 412: A Scaleable OFDMA Engine for WiMAX*
- *AN 421: Accelerating DUC & DDC System Designs for WiMAX*
- *AN 430: OFDMA Ranging for WiMAX*
- *AN 434: Channel Estimation & Equalization for WiMAX*
- *AN 439: Constellation Mapper and Demapper for WiMAX*
- *AN 451: Downlink Subchannelization for WiMAX*
- *AN 452: An OFDM FFT Kernel for WiMAX*
- *AN 457: Integrating Uplink Desubchannelization & Ranging Modules for WiMAX*

Functional Description

Figure 2 shows the uplink desubchannelization reference design block diagram.

Figure 2. Block Diagram



The design has the following four main input and output interfaces:

- Input interface for configuration data
- Input interface for OFDMA symbol data
- Output interface for user slot data
- Output interface for ranging data

The design accepts OFDMA symbol data on the data input interface. The first processing stage is known as the strip. This stage removes the two guard bands that are present around frequency bins $+N/2$ and $-N/2$ for each OFDMA symbol. Also the DC carrier (bin 0) is removed. The remaining frequency bins are written to internal memory. These remaining frequency bins contain ranging data and user data (including pilot information).

The internal memory is double buffered—it can store data for six OFDMA symbols. Thus, while the design processes one group of three symbols and extracts ranging and user data, it reads in another group of the symbols, which undergo the strip process and the design writes them into this internal memory.

User data extraction and ranging data extraction only start to occur after the design writes a group of three consecutive OFDMA symbols into the internal memory buffer, because uplink subchannelization occurs over three consecutive symbols.

Ranging and user data extraction functions effectively occur in parallel.

The ranging data extraction block reads configuration data from the configuration interface to determine the following information:

- The next ranging channel number
- The type of ranging on this ranging channel (initial, periodic, bandwidth, or handover)
- Starting and ending subchannel number for this ranging channel
- The starting slot number for this ranging channel
- The antenna number

Then it determines which subcarriers make up this ranging channel and send requests to the arbiter to read (extract) this data from the internal memory. After it sends the request for the last subcarrier for the current ranging channel, it obtains the next configuration data word for the next ranging channel.

The user slot data extraction block reads configuration data from the configuration interface to determine the following information:

- Single or contiguous range of subchannels to extract
- The starting slot number for this range of subchannels
- The antenna number

Then, for each subchannel, it determines which subcarriers constitute it and sends requests to the arbiter to read (extract) this data from the internal memory. After it sends the request for the last subcarrier for the last subchannel, it obtains the next configuration data word for the next allocation of subchannels.

The arbiter reads the requested data from internal memory and sends it either to the user slot data output interface (if the request originated from user data extraction block) or to the ranging data output interface. If the arbiter receives two requests simultaneously from both the ranging extraction and user data extraction blocks, it services both requests (one after another).

Multiple Antenna Support

This design can be timeshared amongst different antennas. The number of antennas for timesharing depends on the operating clock frequency compared to the required clock frequency to meet throughput.

Each block of three consecutive OFDMA symbols fed into the desubchannelization design must be from the same antenna.

The antenna number must be indicated to the desubchannelization design on both the range map and the user data map input interfaces.

Input Interfaces

The design has the following input interfaces:

- Data bus interface
- Configuration interface:
  - Ranging map interface
  - User data map interface
  - General purpose input (GPI)

All these input interfaces are Altera Avalon® Streaming (Avalon-ST) interfaces (with the exception of the GPI signals that form part of the configuration interface).

For more information on the Avalon-ST interfaces, refer to the *Avalon Streaming Interface Specification*.

Input Interface Signals

Table 1 shows the input interface signals.

All transitions are synchronous to the rising clock edge.

Table 1. Input Interface Signals

Signal	Width	Input /Output	Description
Data Bus Interface			
din_valid	1	Input	Signifies validity of all data bus inputs.
din_ready	1	Output	Signifies whether uplink desubchannelization design can accept more data.
din_data	32	Input	FFT output data (QI samples).
Ranging Map Interface			
rmapin_data	41	Input	Ranging map information.
rmapin_valid	1	Input	Signifies validity of rmapin_data.
rmapin_ready	1	Output	Signifies whether uplink desubchannelization design can accept more rmapin data.
User Data Map Interface			
dmapin_data	35	Input	User data map information.
dmapin_valid	1	Input	Signifies validity of dmapin_data.
dmapin_ready	1	Output	Signifies whether uplink desubchannelization design can accept more dmapin data.

Table 1. Input Interface Signals

Signal	Width	Input /Output	Description
General Purpose Input			
gpin_idcell	7	Input	IDCell value.
gpin_idcell2	7	Input	IDCell value (same as gpin_idcell).

Data Bus Interface

The data bus interface is Avalon-ST compliant (using a ready latency of 1).

Each input sample is the complex frequency domain data representing a single subcarrier. The lowest bits of the data bus are assigned to real portion. A 16-bit width has been assumed for real and also imaginary parts of the sample.

You must feed in the subcarriers for each input OFDMA symbol from frequency bin 0 to bin $N - 1$ (where N is the FFT size).

The data bus interface follows a simple bus protocol. It can accept data samples on consecutive clock cycles.

The upstream agent can increase the interval between feeding in consecutive data samples by holding din_valid low. The validity of all the data bus input signals is indicated by din_valid high. If din_valid is low, the uplink desubchannelization design ignores the data bus inputs.

The design typically reads the data of three consecutive symbols at a time.

It applies backpressure to the upstream agent if it can not process any more data. It applies backpressure by forcing din_ready low. On the next cycle, the upstream agent must force din_valid low and not apply any more valid data until it detects din_ready high.

Ranging Map Interface

The ranging map interface is Avalon-ST compliant (using a ready latency of 1). The protocol is exactly the same as the data bus interface. The uplink desubchannelization design can apply backpressure by deasserting rmapin_ready, forcing the upstream agent to stop sending in more information. Similarly the upstream agent can delay sending new ranging map information by deasserting rmapin_valid.

Table 2 shows the 41-bit range map configuration bus.

Table 2. Range Map Configuration Bus Fields

rmapin_data[40:0] Bits	Field
[1:0]	Header.
[4:2]	Antenna number.
[8:5]	Ranging channel number.
[10:9]	Ranging type.
[17:11]	Starting subchannel number.
[24:18]	Ending subchannel number.
[40:25]	Starting slot number.

Each word on the range map configuration bus refers to one ranging channel.

The antenna number can be a value from 0 to 7. There is support for up to eight antennas.

The ranging channel number can be a value from 0 to 11. For a 2,048 FFT size, there can be a maximum of 11 ranging channels as there are 70 subchannels.

The ranging type provides information on the type of ranging in this channel. Table 3 shows the valid ranging type values.

Table 3. Valid Ranging Type Values

Ranging Type [1:0] Value	Description
00	Initial/handover ranging over two symbols.
01	Initial/handover ranging over four symbols.
10	Periodic/bandwidth ranging over one symbols.
11	Periodic/bandwidth ranging over three symbols.

The *IEEE802.16d/e specifications* state that the ranging channels occupy either four (for a 128 FFT size) or six (for all other FFT sizes) consecutive subchannels. The starting and ending subchannel number provide the range of these four or six subchannels comprising the ranging channel.

The starting slot number is the slot number for the lowest numbered subchannel of the ranging channel.

Each group of three OFDMA symbols requires a packet of range map configuration data. Each packet can comprise from 1 up to 11 different words. The header field of each word can signify the last range map information for the group of three OFDMA symbols in question. If header is equal to three, the current word is the last word in the packet (see Table 4, which shows the decoding of the header word). The next word on this bus is the first word of the next packet, which refers to the next group of three OFDMA symbols.

You must ensure that the antenna number field does not change for each word in a packet.

The uplink desubchannelization design can not begin ranging data extraction until the first word of this configuration packet has been read in. Hence you must ensure that this packet is available at the same time (or before) as each group of three OFDMA symbols is clocked into the design.

Table 4. Range Word Header Field

Header (rmapin_data[1:0]) Value	Header Field
11	Last word in packet.
00,01,10	Not last word in packet.

There is a special case of zero ranging channels. To signify that there are no ranging channels in the group of three OFDMA symbols, a packet of length one must be sent to the range map interface, with the starting subchannel field (`rmapin_data`[29:23]) set to 127 decimal (127 is not a valid subchannel number).

User Data Map Interface

The user data map interface is Avalon-ST compliant (using a ready latency of 1). The protocol of this bus is exactly the same as the data bus interface and ranging map interface.

Table 5 shows the 35-bit user data map configuration bus.

Table 5. User Data Map Configuration Bus Fields (Part 1 of 2)

dmapin_data[34:0] Bits	Field
[1:0]	Header.
[4:2]	Antenna number.

Table 5. User Data Map Configuration Bus Fields (Part 2 of 2)	
dmapin_data[34:0] Bits	**Field**
[11:5]	Starting subchannel number.
[18:12]	Ending subchannel number.
[34:19]	Starting slot number.

Each word on the user data map configuration data bus refers to one range of subchannels allocated to user data. The starting and ending subchannel fields list the range of subchannels (inclusively) that are allocated to user data. If the start subchannel number is equal to the end subchannel number, only one subchannel is defined for user data.

The antenna number can be a value from 0 to 7 (support for up to 8 antennas).

The starting slot number is the slot number for the lowest numbered subchannel in the allocated range.

A packet of user data map configuration data is required for each group of three OFDMA symbols. Each packet can comprise from 1 up to 70 different words. The header field of each word can signify the last user data map information for the group of three OFDMA symbols in question. If header is equal to three, the current word is the last word in the packet (see Table 6, which shows the decoding of the header word). The next word on this bus is the first word of the next packet, which refers to the next group of three OFDMA symbols.

Table 6. User Data Word Header Field	
Header (rmapin_data[1:0]) Value	**Header Field**
11	Last word in packet.
00,01,10	Not last word in packet.

You must ensure that the antenna number field does not change for each word in a packet.

The uplink desubchannelization reference design can not begin user data extraction until the first word of this configuration packet has been read in. Hence you must ensure that this packet is available at the same time (or before) as each group of three OFDMA symbols is being clocked into the design.

You can define contiguous and non-contiguous subchannel ranges for user data allocation. Each word in the packet defines a contiguous subchannel range. The ranges defined in different words may be non-contiguous.

There is a special case of zero subchannels allocated for user data. To signify that there are no subchannels in the group of three OFDMA symbols allocated for user data, a packet of length 1 must be sent to the user data map interface, with the starting subchannel field (`dmapin_data`[29:23]) set to 127 decimal (127 is not a valid subchannel number).

GPI

Two GPIs are available to specify the `IDCell` value for the AP. Both these GPIs, should be driven by the same source and they should always have the same value.

Output Interface Description

The output interface consists of the following two sub-interfaces:

- User slot data interface
- Ranging data interface

All interfaces are Avalon-ST compliant. The behavior of these interfaces is exactly the same as the Avalon-ST input interfaces. For the output interfaces, the uplink desubchannelization design is the source (as opposed to sink), which drives out the data and valid lines, and monitors the ready line.

Output Interface Signals

Table 7 shows the output interface signals.

All transitions are synchronous to the rising clock edge.

Table 7. Output Interface Signals

Signal	Direction	Description
User Slot Data Interface		
`dout_startofpacket`	Output	Start of packet (valid for a single clock cycle high pulse).
`dout_endofpacket`	Output	End of packet (valid for a single clock cycle high pulse).
`dout_data[63:0]`	Output	Packet data.

Table 7. Output Interface Signals

Signal	Direction	Description
dout_valid	Output	The output data is valid.
dout_ready	Input	Indicates whether downstream agent can accept data.
Ranging Data Interface		
drang_startofpacket	Output	Start of packet (single clock cycle high pulse).
drang_endofpacket	Output	End of packet (single clock cycle high pulse).
drang_data[63:0]	Output	Packet data.
drang_valid	Output	The output data is valid.
drang_ready	Input	Indicates whether downstream agent can accept data.

You can configure the output data bus widths at synthesis time. This reference design assumes 16-bit wide I and Q samples.

User Slot Data Output Interface

This interface is Avalon-ST compliant (using a ready latency of 1). The downstream agent can apply backpressure to the uplink desubchannelization design if it cannot accept the data by deasserting dout_ready.

The interface outputs packets of data, with each packet marked by all valid data words output when a high is detected on dout_startofpacket until a high is detected on dout_endofpacket i.e. dout_startofpacket = 1 for the first sample and dout_endofpacket = 1 for the last sample in packet. Each packet comprises a slot (WiMAX terminology) of user data.

A slot comprises a subchannel over three consecutive OFDMA symbols. There are six tiles (where a tile is four consecutive subcarriers) in each symbol allocated to a subchannel. Over three symbols this equates to 72 samples (6 × 4 × 3). A subchannel has only 48 samples. The remaining samples are pilot signals. All 72 samples for each valid slot are output. The pilot samples are required for the channel estimation function that follows this block.

The samples in each slot are output in the following order:

- lowest numbered tile tn in the first symbol
- tile tn in symbol 2
- tile tn in symbol 3
- tile tn + 1 in symbol 1

- tile tn + 1 in symbol 2...
- tile tn + 5 in symbol 3

Each output data word contains the complex subcarrier data plus some sideband information, some of which may be required for downstream processing blocks (channel estimation, carrier frequency offset estimation, and correction). Table 8 shows the output user data slot fields.

Table 8. Output User Data Slot Fields

dout_data[63:0] Bits	Field
[63:61]	Antenna number.
[60:53]	Lowest eight bits of slot number.
[52:51]	Symbol offset.
[50:44]	Subchannel number (0..69).
[43:35]	Physical tile number (0..419).
[34:32]	Logical tile index (0..5).
[31:16]	Imaginary part of complex subcarrier value.
[15:0]	Real part of complex subcarrier value.

Ranging Data Output Interface

This interface is Avalon-ST compliant (using a ready latency of 1). The downstream agent can apply backpressure to the uplink desubchannelization design if it cannot accept the data by deasserting drang_ready.

The interface outputs packets of data, with each packet marked by all valid data words output when a high is detected on drang_startofpacket, until a high is detected on drang_endofpacket (drang_startofpacket = 1) for the first sample and drang_endofpacket = 1 for the last sample in packet. Each packet comprises a slot (WiMAX terminology) of user data.

A ranging channel in one OFDMA symbol (either 144 or 96 samples) constitutes one packet on this output.

The ranging extraction begins from the first listed ranging channel in the input configuration map read in, to the last listed ranging channel. A ranging channel comprises either four subchannels (for a 128 FFT size) or six (for all other FFT sizes) subchannels. A subchannel comprises six tiles in one symbol. All subcarriers within a tile are used for ranging data

(there are no pilot signals). Thus a ranging channel equates to either 96 samples (for a 128 FFT size) or 144 samples (for all other FFT sizes) per OFDMA symbol.

For multiple ranging channels, all the ranging data for one ranging channel over all three symbols is output before moving onto the next ranging channel. For each ranging channel the samples are output in order from lowest numbered tile in first symbol to highest numbered tile in first symbol, then moving onto symbol 2 and then onto symbol 3 in the same order. With the ranging data, sideband information is output.

The complex ranging data along with this sideband information is output on `drang_data`. Table 9 shows the bit mapping of this word to the different fields. The sideband information remains the same for the entire packet.

Table 9. Ranging Output Word

drang_data[63:0] bits	Field
[63:57]	First subchannel number (next 5/3 also comprise ranging channel).
[56:55]	Symbol offset (0..2).
[54:51]	Ranging channel number.
[50:49]	Ranging type.
[48:35]	Lowest 14 bits of starting slot number for ranging channel.
[34:32]	Antenna number.
[31:16]	Imaginary part of ranging data.
[15:0]	Real part of ranging data.

Getting Started

This section describes the system requirements, installation and other information about using the uplink desubchannelization reference design.

System Requirements

The reference design requires the following hardware and software:

- A PC running the Windows 2000/XP operating system
- Quartus® II software version 6.0, SP1
- ModelSim SE 5.7d (mixed VHDL-Verilog HDL license)

Install the Reference Design

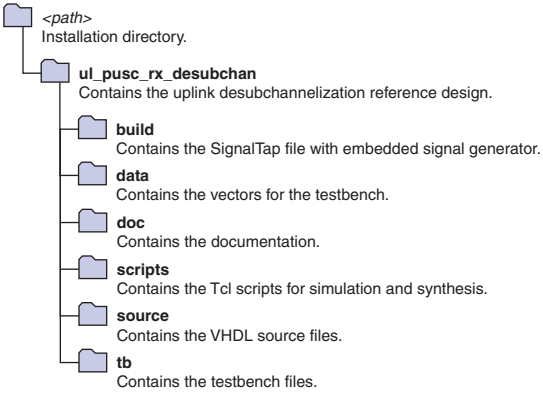
The uplink desubchannelization reference design ships with the scaleable OFDMA engine.

For more information and installation instructions on the scaleable OFDMA engine, refer to *AN412: A Scaleable OFDMA Engine for WiMAX*.

The reference design installs by default into the **c:\altera\reference_designs** directory, but you can change the default.

Figure 3 shows the directory structure, where <*path*> is the top-level directory, **wimax_ofdma\source\rtl\ul_rx**.

Figure 3. Directory



After you install the reference design, follow these steps:

1. Browse to the <*Quartus II install directory*>**\libraries\vhdl\altera**.
2. Make a backup copy of the existing **alt_cusp_package.vhd** file.
3. Copy the **alt_cusp_package.vhd** file from**\source\ul__pusc_rx_process_128\dump**directory and paste to the <*Quartus II install directory*>**\libraries\vhdl\altera**.

Table 10 lists the files associated with test pattern generation and hardware debug of the reference design.

Table 10. Debug Files

File Name	Directory	Information
ul_desub_debug_tb.vhd	**\tb**	RTL testbench for test pattern test.
ul_desub_debug_toplevel.vhd		
ul_desub_debug.vhd		
debug_din_if.vhd		
debug_config_dmap_if.vhd		
debug_config_rmap_if.vhd		
ul_desub_debug_pkg_n2048.vhd		2K FFT test package file.
ul_desub_debug_pkg_n1024.vhd		1K FFT test package file.
ul_desub_debug_pkg_n512.vhd		512 FFT test package file.
ul_desub_debug_pkg_n128.vhd		128 FFT test package file.
ul_desub_debug_quartus.tcl	**\scripts**	Quartus script to build design with embedded test pattern generation.
ul_desub_debug_tb_msim.tcl	**\scripts**	ModelSim script for embedded test pattern simulation.
stp1.stp	**\build**	SignalTap® file for design with embedded test pattern generation.

Understand the Data Files

Altera provides the following data files in **\wimax_ofdma\source\rtl\ul_rx\ul_pusc_rx_desubchan\data**:

- Input data: **ip_data_desub_range_**<*identifier*>**.txt**
- Input range map configuration data: **rmapip_data_desub_range_**<*identifier*>**.txt**
- Input user data map configuration data: **dmapip__data_desub_range_**<*identifier*>**.txt**
- Output user slot data: **op_data_desub_range_**<*identifier*>**.txt**

Altera provide ranging output data from the uplink desubchannelization design in **\wimax_ofdma\source\rtl\ul_rx\ranging_ap\data** as output ranging data: **ipdata_desub_range_**<*identifier*>**.txt**.

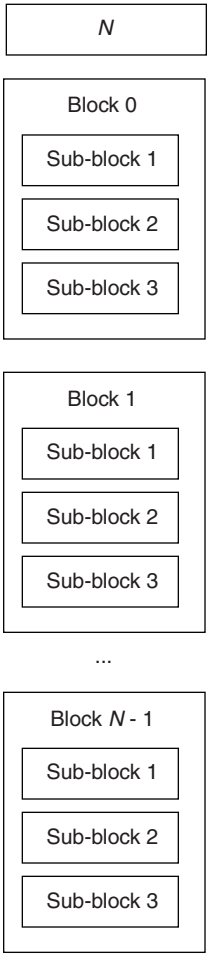
You can use these data files in an appropriate RTL testbench.

Input & Output Data File Formats

This section describes the format of the input and output data files.

Figure 4 shows the general structure of these files. The data is arranged into several blocks. Each block has three sub-blocks. All fields must contain decimal numbers only.

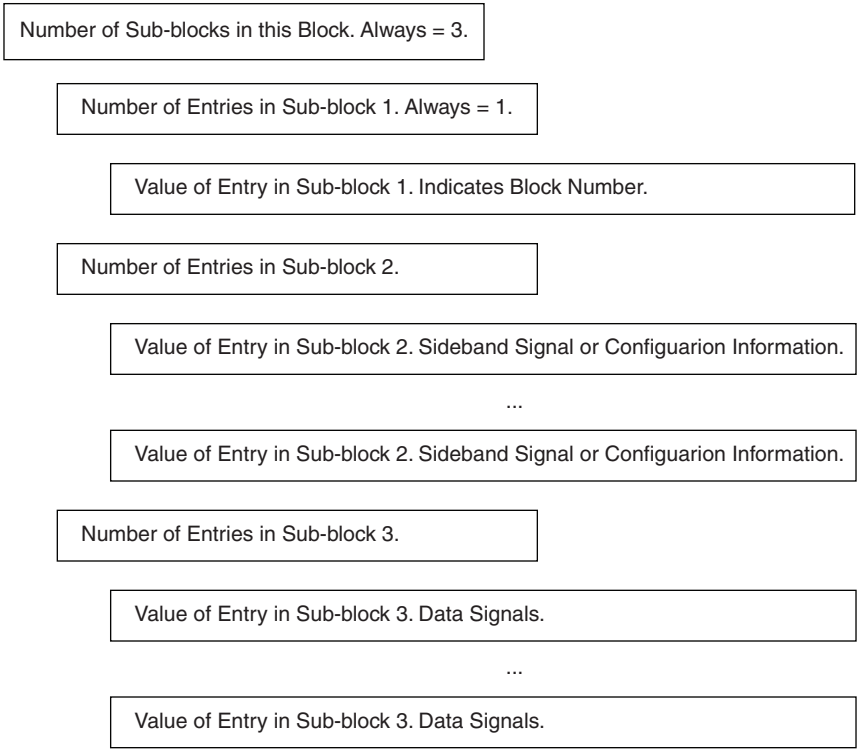
Figure 4. General Data Text File Structure



The first line contains a single number *N*. This number indicates the number of blocks in the file. If the number is zero, ignore its value, as the number of blocks in the file is unknown or not calculated.

After the first line, each block of data follows. Each block comprises three sub-blocks. Figure 5 shows the structure of each block in more detail. The first line in each block contains a single number, which indicates the number of sub-blocks in this current block and should always be 3.

Figure 5. Structure of Single Block in File



The next line contains a single number that indicates the number of entries in sub-block 1, which should always be 1. The following line contains a single number, which is the value of the entry for sub-block 1. Sub-block 1 only contains a block number ID, given by this value.

The next line contains a single number that indicates the number of entries in sub-block 2. Sub-block 2 contains all sideband signal and configuration information. The number of entries in this sub-block varies, depending on which file is being referred to. For example for 12 entries, the next 12 lines in the files contain the sideband signal and configuration information.

The next line contains a single number, *m* that indicates the number of entries in sub-block 3. The next *m* lines contain information on the data signal values.

Sub-Block 2: Sideband Information

The following code is an example of the possible contents of sub-block 2:

```
10
17 117503728
16 4
11 2
12 4
18 144
19 0
20 8
21 8
22 8
23 8
```

The first line contains 10, which means that there are 10 entries in sub-block 2. The next 10 lines contain the sideband signal and configuration information. There are 2 numbers on each line. The first number is a code that indicates which sideband signal or configuration information is referred to. The second number gives the value for this sideband signal or configuration information. Table 11 shows the sub-block 2 field codes.

Table 11. Sub-Block 2 Field Codes (Part 1 of 2)

Code	Signal	Present in Text File				
		Input Data	Range Map Input Data	Data Map Input Data	Output Slot Data	Output Ranging Data
29	Antenna number		Yes	Yes	Yes	Yes
0	Ranging channel number					Yes
1	Ranging type					Yes
2	Starting slot number		Yes			Yes
19	Start Range Code Offset		Yes *(1)*			
20	Number of initial ranging codes		Yes *(1)*			
21	Number of periodic ranging codes		Yes *(1)*			
22	Number of bandwidth ranging codes		Yes *(1)*			
23	Number of handover ranging codes		Yes *(1)*			
11	FFT size code		Yes	Yes		Yes

Table 11. Sub-Block 2 Field Codes (Part 2 of 2)

Code	Signal	Present in Text File				
		Input Data	Range Map Input Data	Data Map Input Data	Output Slot Data	Output Ranging Data
12	Cyclic prefix length code		Yes	Yes		Yes
24	Max Positive Timing Offset					Yes *(1)*
25	Max Negative Timing Offset					Yes *(1)*
26	Margin Positive Timing Offset					Yes *(1)*
27	Margin Negative Timing Offset					Yes *(1)*
14	Code detection threshold					Yes *(1)*
15	Symbol Offset					Yes
16	`IDCell` of basestation	Yes	Yes	Yes	Yes	Yes
17*	MATLAB initial random generator seed	Yes *(1)*	Yes *(1)*	Yes *(1)*	Yes *(1)*	Yes *(1)*
506	Subchannel Number				Yes	Yes
500	3 consecutive symbol counter	Yes *(1)*				

Note to Table 11:

(1) Extra signals that provide more information about test setup, which may be useful when integrating with downstream and upstream blocks.

Sub-Block 3: Data Values

This section describes the sub-block 3 data values.

Input Data File

The input file is arranged so that each block represents data for a particular OFDMA symbol.

The following code shows an example of the start of sub-block 3:

```
1024
9631 8483
12116 -14994
-22526 -3079
-8978 10877
-17961 1604
-2809 -4338
-7953 18214
.............
```

The first line contains the number of data samples in an OFDMA symbol, which is 1,024 (the FFT size is equal to 1,024 for this example). This value is equal to the FFT size (128, 512, 1,024 or 2,048). Each line contains two numbers. The first is the real part of the sample; the second is the imaginary part.

The subcarriers for each input OFDMA symbol are listed, in order from frequency bin 0 to bin $N - 1$ (where N is the FFT size).

Input Ranging Map File

The third sub-block in this file represents the map for how many ranging channels and which subchannels are allocated to ranging for a group of three consecutive OFDMA symbols.

The following code shows an example of sub-block 3:

```
2
0 0 0 0 5 28644
3 1 2 6 11 28650
```

The first line contains the number of lines in this sub-block. This equates to the number of different ranging channels (two in this case).

Each subsequent line refers to one ranging channel and contains six fields. From left to right each field represents:

```
Field 1: Header
Field 2: Ranging Channel Number
Field 3: Ranging Type
Field 4: Starting subchannel number allocated to this
ranging channel
Field 5: Last subchannel number allocated to this
ranging channel
Field 6: Starting Slot number for lowest numbered
subchannel allocated to this ranging channel
```

Input User Data Map File

The third sub-block in this file represents the map for which subchannels are allocated to User data for a group of three consecutive OFDMA symbols.

The following code shows an example of sub-block 3:

```
4
0 7 10 3235
1 20 26 3248
2 6 6 3234
3 30 32 3258
```

The first line indicates that there are four different subchannel regions allocated to user data. The next four lines define each of these subchannel regions.

Each subsequent line contains four fields. From left to right each field represents:

```
Field 1: Header
Field 2: Starting Subchannel number allocated to user
data
Field 3: Last Subchannel number allocated to user data
in this region
Field 4: Slot number for first subchannel number in
this region
```

Output User Slot Data

Sub-block 3 represents the user data for a particular slot (one subchannel over three consecutive OFDMA symbols).

The following code shows an example of the possible contents of sub-block 3:

```
72
28185 7243 0 0
-7939 3651 0 0
23976 -11608 0 0
-14223 14432 0 0
6219 7954 1 0
-2072 11659 1 0
-14432 -2320 1 0
-15028 -18796 1 0
-27025 11739 2 0
25039 -1896 2 0
...
```

The first line indicates that there are 72 subsequent lines in this sub-block; this equates to the number of data samples in one user data slot.

Each subsequent line contains information about each slot sample. Each line always contains four fields. Going from left to right, the different fields represent:

```
Field 1: Real part of slot data
Field 2: Imaginary part of slot data
Field 3: OFDMA Symbol offset (0..2) within group of
three symbols
Field 4: Logical tile index that slot data refers to
(value from 0...5)
```

Output Ranging Data
Sub-block 3 represents one ranging channel data for a particular OFDMA symbol.

The following code shows an example of the start of sub-block 3:

```
144
17388 23877
21709 19399
16389 28867
6923 26390
-14143 -9200
-13723 -17297
13323 17627
12076 17699
.............
```

The first line contains the number of data samples in a ranging channel, which are 144 in this example. This value can be either 144 or 96 (when FFT size is 128). The next 144 (or 96) lines contain the ranging channel data.

Each line contains two numbers. The first is the real part of the sample, and the second is the imaginary part. The samples are ordered from the lowest numbered subcarrier of the lowest numbered subchannel that comprises a ranging channel to the highest numbered subcarrier of the highest numbered subchannel of the ranging channel.

Debug the Reference Design

To understand the subcarriers that are extracted, feed a repeatable test pattern into the desubchannelization design, which has an unique value for each different subcarrier.

This method can also confirm that the uplink subchannelization block in the SS is basing its mapping on the same one as this desubchannelization block (to be used in basestation).

The following files enable you to perform this debug:

- Test pattern generation logic
- Top-level design file that instantiates test logic and desubchannelization design
- RTL testbench
- Modelsim RTL simulation script
- Script that synthesizes test logic with desubchannelization design for the Stratix® II DSP development board

You can load the test pattern logic into the hardware, which allows real hardware debug using either logic analyzers or Altera's SignalTap. For demonstration purposes, scripts synthesize the design for the Stratix II DSP development board and allow you to use SignalTap to view outputs from the uplink desubchannelization design.

Test Pattern

The generated test pattern is different for the real and imaginary parts of the data that is fed into the design.

Real Data Input
The design processes OFDMA symbols in groups of three.

The real data input, (`din_real`[15:0]), has an unique value for each subcarrier for each OFDMA symbol in a block of three OFDMA symbols. Then the values repeat for each block of three OFDMA symbols.

The real data is subdivided into several signals (see Table 12) and a test pattern generated for each of these signals.

Table 12. Real Data Test Signals

Din_real[15:0] Bit Range	Test Signal Field Name
[15:14]	`T_symbol_no`
[13:0]	`T_subcarrier_code`
[13:12]	`T_subcarrier_type`
[11:0]	`T_subcarrier_subcode`

Tables 13 through Table 15 show the possible values and meanings of these signals.

Table 13. Encoding of T_symbol_no

T_symbol_no[1:0] Value	Occurrence
00	First OFDMA symbol in a block of three.
01	Second OFDMA symbol in a block of three.
10	Third OFDMA symbol in a block of three.

Table 14. Encoding of T_subcarrier_type

T_subcarrier_type[1:0] Value	Occurrence
00	Usable Subcarrier.
01	Left guard band subcarrier.
10	Right guard band subcarrier.
11	DC carrier.

Table 15. Encoding of T_subcarrier_subcode

T_subcarrier_subcode[11:0] Value	Occurrence
0	DC, or first left guard, or first right guard, or first usable subcarrier (determined by `T_subcarrier_type`).
n	The *n*th left guard, or *n*th right guard, or *n*th usable subcarrier (determined by `T_subcarrier_type`).

Note to Table 15:
(1) The usable subcarrier numbering of 0 to *n* is according to the WiMAX specifications.

Table 16 shows the test sequence for a 1,024K FFT size. Other FFT sizes have similar patterns. The only differences are the number of guard and usable subcarriers.

Table 16. Example Test Pattern on Real Data for 1,024 FFT Size (Part 1 of 2)

Sample Number	Freq Bin	T_symbol_ no[1:0]	T_subcarrier_ type [1:0]		T_subcarrier_no [11:0]	T_subcarrier_code [13:0]
1	0	0,1 or 2	DC subcarrier	3	0	12,288 (14'h3000)
2	1	0,1 or 2	Usable subcarrier	0	420	420 (14'h01A4)
3	2	0,1 or 2	Usable subcarrier	0	421	421 (14'h01A5)
...	...	...	...	...	...	...
421	420	0,1 or 2	Usable subcarrier	0	839	839 (14'h0347)
422	421	0,1 or 2	Right guard	2	0	8,192 (14'h2000)

Table 16. Example Test Pattern on Real Data for 1,024 FFT Size (Part 2 of 2)

Sample Number	Freq Bin	T_symbol_ no[1:0]	T_subcarrier_ type [1:0]		T_subcarrier_no [11:0]	T_subcarrier_code [13:0]
423	422	0,1 or 2	Right guard	2	1	8,193 (14'h2001)
...	...	...	...	...	...	...
512	511	0,1 or 2	Right guard	2	90	8,282 (14'h205A)
513	–512	0,1 or 2	Left guard	1	0	4,096 (14'h1000)
514	–511	0,1 or 2	Left guard	1	1	4,097 (14'h1001)
...	...	...	...	...	...	...
604	–421	0,1 or 2	Left guard	1	91	,4187 (14'h105B)
605	–420	0,1 or 2	Usable subcarrier	0	0	0 (14'h0000)
606	–419	0,1 or 2	Usable subcarrier	0	1	1 (14'h0001).
...	...	...	...	...	...	...
1,024	–1	0,1 or 2	Usable subcarrier	0	419	419 (14'h01A3)

The desubchannelization block expects data to be fed into it in order from frequency bin DC onwards.

Thus, referring to FFT size of 1,024 (for example) and Table 16, the test pattern is generated in order from frequency bins 0 to 511, then –512 through to –1.

Imaginary Data Input

The imaginary data input is the output from a 16-bit counter, which increments every cycle.

This input verifies that on the output the same subcarriers from different groups of three OFDMA symbols are from different groups of three OFDMA symbols and not the same group (hardware is processing more than 1 group of three OFDMA symbols). The real data has the same value but the imaginary data has a different value.

Configuration Data Map In

The same configuration data map is sent to the design, whenever the design indicates it is ready to accept another configuration data word.

The configuration packet consists of one word only. Thus only one subchannel range can be defined.

The subchannels to be extracted for user data are set in the package files (one package file for each different FFT size). You can overwrite the default settings in here.

Configuration Range Map In
The same configuration range map is sent to the design, whenever the design indicates it is ready to accept another configuration range word.

The configuration packet consists of one word only; so up to one ranging channel can be defined. The subchannels that comprise this channel are set in the package files (one package file for each different FFT size). You can overwrite the default settings in here.

RTL Simulation of Desubchannelization with Test Pattern

Altera provides an RTL testbench with a Tcl script (**ul_desub_debug_tb_msim.tcl**) that can be run from the Modelsim simulator.

When you run the Tcl script, it performs the following actions:

- Compiles the uplink desubchannelization reference design
- Compile the test pattern generation logic
- Loads up of waveform viewer

The testbench feeds a clock and reset signals into the system (desubchannelization plus test pattern generation logic) and writes the user data and ranging outputs from the desubchannelization to the following text files respectively:

- **sim\desub_op_ddata.txt**
- **sim\desub_op_rdata.txt**

The simulation runs indefinitely, as data is repeatedly fed into the desubchannelization design. Thus, you must stop the simulation.

You can simulate all desubchannelization design versions for the different FFT sizes.

To specify the FFT size modify the following variables at top of Tcl script:

- `set fftsize 1024`
- `set proj_topdir "D:/work/WiMax/wimax_ofdma/source/rtl/ul_rx/ul_pusc_rx_desubchan"`

The first variable sets the FFT size. 128, 512, 1024, and 2048 are valid values; the second variable is the path to the top-level file of the desubchannelization design.

After, modifying these two variables, run the script in the ModelSim simulator.

Hardware Simulation using the Stratix II Development Board

You can load the uplink desubchannelization reference design and the test pattern generation logic onto an FPGA to test on hardware.

The **ul_desub_debug_toplevel.vhd** file is the top-level file that instantiates both the design and the test pattern generation logic (the debug system).

The Tcl script **ul_desub_debug_quartus.tcl** synthesizes the debug system for Altera's Stratix II DSP Development Board and enables SignalTap on the design's outputs.

To perform hardware simulation, follow these steps:

1. Edit following lines in **ul_desub_debug_quartus.tcl**, to reflect the FFT size to synthesize, the location of the design, and the version of the Quartus II software.

```
set fftsize 128
set proj_topdir
"D:/work/WiMax/wimax_ofdma/source/rtl/ul_rx/ul_pusc_r
x_desubchan"
# set to one if Quartus S/W is earlier than Quartus 6.1
set pre_quartus61 1
```

2. In the Quartus II software change the directory to location of **ul_desub_debug_quartus.tcl**.

3. In the Quartus II Tcl console type the following command:

```
source ul_desub_debug_quartus.tcl
```

   The debug system is synthesized for a Stratix II 2S60 device to generate following file:

build/ul_desub_debug_toplevel.sof

4. Download this **.sof** file to the Stratix II 2S60 device on the development board using the programming cable.

5. On the Tools menu click **SignalTap Logic Analyzer** and click **Acquire data**.

A waveform of the outputs from the desubchannelization block displays.

Performance

This section shows the synthesis results and throughput.

Synthesis Results

Table 17 shows the synthesis results for all FFT sizes. The results assume 16-bit inputs for the real and imaginary parts of the input data that is fed into the design and the slot data and ranging data output.

Table 17. Uplink Desubchannelization Synthesis Results

Device	FFT Size	LEs/ALUTs	Memory			9 × 9 Multipliers	F_{MAX} (MHz)
			M512	M4K	MRAM		
Cyclone II 2C35 C6	128	3,521 (11%)	–	20 (19%)	–	2 (3%)	159
	512	3,523 (11%)	–	44 (42%)	–	2 (3%)	153
	1,024	3,461 (10%)	–	52 (50%)	–	2 (3%)	147
	2,048	3,518 (11%)	–	92 (88%)	–	2 (3%)	143
Stratix II 2S30 C4	128	3,179 (12%)	2 (<1%)	18 (13%)	–	2 (2%)	180
	512	3,235 (12%)	2 (<1%)	42 (29%)	–	2 (2%)	176
	1,024	3,180 (12%)	2 (<1%)	10 (7%)	1 (100%)	2 (2%)	168
	2,048	3,226 (12%)	2 (<1%)	10 (7%)	1 (100%)	2 (2%)	180

For Cyclone® II synthesis, you need to set `VERIFIED_SAFE` setting for synthesis parameter `CYCLONEII_SAFE_WRITE` under **Default Parameters** of Analysis & Synthesis Settings. Otherwise, you see twice the increase in M4K block usage. This workaround is for a silicon issue, when using memory in certain configurations. However, this design does not use the memory in any of the configurations where the silicon issue is a problem.

For more information, refer to the *Cyclone II Errata Sheet*.

Throughput

OFDMA symbols are continuously fed into the uplink desubchannelization reference design at a data rate appropriate to the FFT size. The design must be able to output all ranging and user data for a group of three OFDMA symbols within the time taken for the next three OFDMA symbols to be fed into the block.

To meet throughput ,the design must be clocked at a minimum of four times the data rate for all FFT sizes.

Table 18 shows the minimum clock frequency for throughput.

Table 18. Minimum Clock Frequency

FFT Size	Data Rate (MHz)	Minimum Clock Frequency (MHz)
128	1.25	5
512	5	20
1,024	10	40
2,048	20	80

Revision History

Table 19 shows the revision history for this application note.

Table 19. Revision History

Version	Date	Description
1.0	February 2007	First release.

101 Innovation Drive
San Jose, CA 95134
(408) 544-7000
www.altera.com
Applications Hotline:
(800) 800-EPLD
Literature Services:
literature@altera.com