SIMULATION HELP

Project:	TestiPHY
Author:	Neil Shipp
Date:	19 December 2002
Title:	TestiPHY 802.16a OFDM Simulation v1.2 Help
Reference:	TestiPHY/DOC/WORKING/W00058/1.2
Distribution:	

1. INTRODUCTION

802.16a is the IEEE Standard for Broadband Wireless Access in the bands 2-11GHz. This standard has 3 options for a Physical Layer interface and this simulation is concerned with the OFDM option, sometimes termed the WirelessMAN OFDM PHY. This simulation has been written to comply with version D7 – which it is hoped will become the final version of the standard.

More information on the 802.16 standard, including tutorials, may be found on the working groups web site at http://www.wirelessman.org/

The standard does not define any particular bandwidth other than a minimum. It is intended that it should be targeted at whatever regulatory bandwidths are available and sub-multiples thereof. For example allocations within the 5GHz UNII band are typically 20MHz, whereas the 10GHz licensed bands are 28MHz.

1.1 Simulation Purpose

This simulation is intended as a tool to help resolve issues with parts of the specification that may be open to multiple interpretations. For a Standard to have real value then equipment from one vendor must be able to interwork with that of another. It is hoped that most of these issues can be resolved at the simulation stage, at least for the PHY layer, and avoid some of the problems that have held back other standards.

The simulation represents my current interpretation of the Standard – if yours is different please voice your comments to me at mailto:nrs@commsonic.com or via the 802.16 email reflector.

1.2 Simulation Limitations

The simulation will generate single bursts of data, formatted for downlink or uplink in any of the mandatory coding schemes and modes. The optional block and convolutional turbo coding FEC modes are not supported.

The modulator is currently limited to producing single bursts of data – or in the case of the downlink an FCH burst followed by a single burst of data.

The demodulator is greatly simplified and is there simply to synchronise to the OFDM symbols, provide a constellation diagram and make the simulation a bit more interesting.

1.3 Installation

This simulation is designed to run on the PC Windows version of Matlab and the Signal Processing Toolbox. It has been tested on both version R12 and R12.1. It should also work correctly on the latest version (R13) but this has not yet been tested.

Installation Instructions:

- Put the zip file in your project folder.
- Using Winzip or Pkzip extract into this folder with the 'use folder names' option selected.
- You should see a Matlab icon in your project folder which you can then double click to launch Matlab - it is important that you start Matlab from this as it sets all the paths up correctly. If this link is not set correctly then right click on it, select Properties and set the path to your Matlab executable - something like C:\matlabR12\bin\win32\matlab.exe and also set the 'Start in:' to blank.
- When Matlab starts the GUI should start too. Select whatever options you require and press RUN.

1.4 Key 802.16a OFDM Parameters

- COFDM based on a 256 point FFT, central 200 carriers used and DC nulled.
- 8 carriers are used as pilot tones.
- Cyclic prefix (or Guard Interval) of 1/32, 1/16, 1/8 or 1/4 symbol period.
- ¼ band and ½ band sub-channels also defined
- QPSK, 16QAM or 64QAM Data Carriers
- Concatenated Reed-Solomon and Convolutional FEC with interleaving.

2. SIMULATION PARAMETERS

2.1 Filenames Menu

2.1.1 Input Source

There are 3 options to determine the input data stream to the modulator.

- Random – Random data generated of length determined by the 'length' edit box
- File – Data will be read from the file selected by 'Input Filename'. The format of this file should be CRLF delimited ASCII byte data. An example is given in the 'work' folder as 'datain.txt'.
- Test Pattern – This uses a 35 byte test pattern currently used in the standard to provide an example of coding in one particular Uplink mode.

2.1.1.1 Input Length

This is the length of the input data in bytes. It is only effective for a random data input otherwise the data length will be determined by the length of the input file. An arbitrary limit of 4095 bytes has been set.

2.1.2 Debug Enable

This enables a debug dump of intermediate data within the modulator. Data is logged at the following points:

- Input
- Scrambler Output
- Reed-Solomon Output
- Convolutional Encoder Output
- Interleaver Output
- Mapped Data Output

2.1.3 Input Filename

Opens a standard Matlab file selection dialog box. Input file should be CRLF delimited ASCII byte data. An example 'datain.txt' is provide in the /work folder.

2.1.4 Output Filename

Opens a standard Matlab file selection dialog box. Output file will be written in a proprietary Binary Complex Data (*.bcd) format. This format is written by the Matlab command:

```
fwrite(opfid, [real(TxData); imag(TxData)], 'float32');
```

The included utility 'bcdread.m' may be used to read a *.bcd file back and shows an example of the code required to read the file and convert it back to a complex double array. Default output file will be 'txdata.bcd'

2.1.5 Debug Filename

Opens a standard Matlab file selection dialog box. Output will be written as an ASCII text file. The file may be viewed using Notepad or any other text viewer. Default output file will be 'debugdata.txt'

2.2 Framing Menu

2.2.1 Uplink/Downlink

Sets the direction of the link. Uplink means a transmission from the Subscriber Station to the Basestation and Downlink is the reverse. This changes a few system parameters – most significantly the preambles used and reset values of data and pilot scramblers.

2.2.2 SubChannel Enable

Subchannelisation may only be enabled when running the system as an Uplink. The Subchannel index setting specifies the subchannel format to be used.

2.2.3 IUC

For the Downlink the Interval Usage Code determines where a burst has come from in a frame sequence. This simulation currently only encodes single bursts so this value has no effect.

For the Uplink the Interval Usage Code is encoded in the reset value of the scrambler.

2.2.4 Slot

For the Uplink this is the OFDM symbol number within the frame that the Uplink burst starts. It is also encoded in the reset value of the scrambler.

2.3 Help Menu

Takes you to this information, software revision information.

2.4 Main Window

2.4.1 Modulator

2.4.1.1 Rate ID

The 802.16a specification currently defines 6 Rate ID encodings. Select the different ID's to see the parameters summarised in the adjacent text window. e.g RateID 0:

Modulation: QPSK
Coded Block Size: 48 bytes
Uncoded Block Size: 24 bytes
Overall Code Rate: ½
Reed-Solomon Parameters: N=32, K=24, T=4

2.4.1.2 SubChannel Index

SubChannelisation may only be selected when the system is run as an Uplink. Subchannel indices 1 to 4 select one of the ¼ band subchannels. Indices 5 and 6 select one of the ½ band subchannels.

2.4.1.3 Guard

This selects the length of the Guard Interval or Cyclic Prefix as it is otherwise known. Guard intervals of 1/32, 1/16, 1/8 and 1/4 symbol are valid.

2.4.2 Channel

2.4.2.1 SNR

Set level of AWGN (Additive White Gaussian Noise) as SNR in dB. This assumes the bandwidth is the full 201 carriers of the 256 point FFT – hence individual carrier SNR will appear better when sub-banding is used.

2.4.3 Demodulator

2.4.3.1 Display

There are 3 display options:

- Constellation: An I/Q plot of the received constellation. The BPSK pilot tones have been coloured differently to the data carriers. In downlink mode the FCH QPSK data is coloured red.

- Correlator: Shows the output of a correlation with the known long preamble sequence. Two strong correlation spikes should be visible.

- Spectrum: Power spectral density plot of the modulator output.

3. MODULATOR DESCRIPTION

The following text gives a description of the core modulator script 'mod16ofdm.m'.

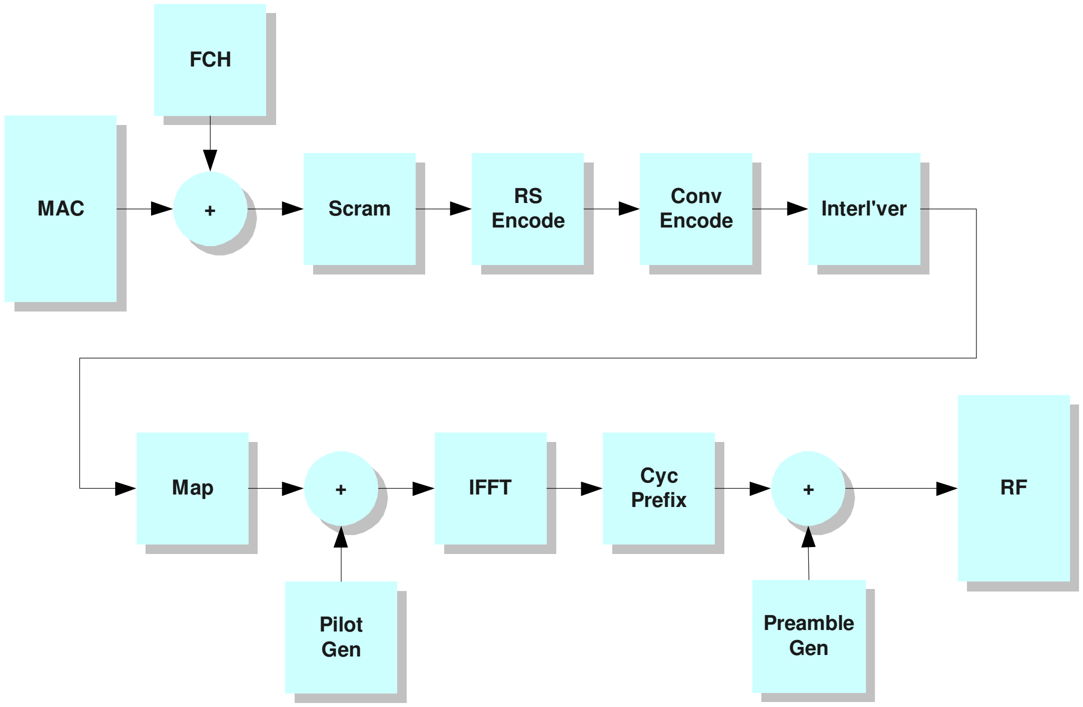

Figure 1: Modulator Block Diagram

Line 1 :165

These lines set up various system parameters. Most of the case statements are concerned with converting the 'RateID' and Subchannel Index parameters into the various combinations of QAM level, Block Size and FEC parameters.

Line 165:180

Load the different Data and Pilot carrier locations and preamble sequences for normal or sub-channel modes. Assemble the preamble sequences and write to file.

Line 185:195

Pads the data to an integer number of OFDM symbols. Padding is done with 0xFF bytes except the final byte which must be left for an 0x00 to be appended to flush the convolutional encoder.

Line 195:205

Calculate the pilot scrambling sequences

Line 210:290

A FCH (Frame Control Header) burst is the first burst transmitted in Downlink mode. The basic content of this symbol is the Downlink frame prefix which is implemented in this simulation. We have interpreted the HCS generation to be a Hamming type polynomial division, where the transmitter takes the RateID and Length bytes, divides by the generator polynomial and uses the remainder as the HCS code.

Our interpretation is that the FCH is padded with 0xFF bytes and a final 0x00. It is not scrambled but passed through the standard ½ rate encoding process for QPSK.

Line 290:305

Scramble the data with the appropriate LFSR initialisation depending n whether an Uplink or Downlink. Append a 0x00 byte at the end of the data to flush the convolutional encoder.

Line 305:330

Reed-Solomon encoding using a N=255, K=239, T=8 encoder. 802.16a uses much smaller code blocks based on this RS code by puncturing the large code blocks down to the required size and so the data is prepended with zeros and unwanted parity bytes and output zeros are truncated. The standard requires the parity bytes to be transmitted first. Finally convert to bit stream data for the convolutional encoder.

Line 330:340

Constraint length 7 convolutional encoder. Now with zero initialisation and zero flushing on each burst.

Line 345:350

Two stage interleaver. The first stage permutes the individual bits. The second stage separates similarly sensitive bits from the QAM constellation. The second stage does nothing in QPSK mode.

Line 350:360

Map the bits to FFT carriers. Interleave pilot tones and a DC carrier into the data carriers and calculate the scrambled pilot tone values.

Line 365:375

Expand the carriers to 256 by adding the appropriate number of null carriers above and below. FFT the data and add a cyclic prefix.

Line 375:400

Provide debug data for inter-block signals.