Notice

The information contained in this document is subject to change without notice.

Agilent Technologies makes no warranty of any kind with regard to this material, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. Agilent Technologies shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance, or use of this material.

Warranty

A copy of the specific warranty terms that apply to this software product is available upon request from your Agilent Technologies representative.

Restricted Rights Legend

Use, duplication or disclosure by the U. S. Government is subject to restrictions as set forth in subparagraph (c) (1) (ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013 for DoD agencies, and subparagraphs (c) (1) and (c) (2) of the Commercial Computer Software Restricted Rights clause at FAR 52.227-19 for other agencies.

Agilent Technologies
395 Page Mill Road
Palo Alto, CA 94304 U.S.A.

Copyright © 1998-2004, Agilent Technologies. All Rights Reserved.

Acknowledgments

Mentor Graphics is a trademark of Mentor Graphics Corporation in the U.S. and other countries.

Microsoft®, Windows®, MS Windows®, Windows NT®, and MS-DOS® are U.S. registered trademarks of Microsoft Corporation.

Pentium® is a U.S. registered trademark of Intel Corporation.

PostScript® and Acrobat® are trademarks of Adobe Systems Incorporated.

UNIX® is a registered trademark of the Open Group.

Java™ is a U.S. trademark of Sun Microsystems, Inc.
## Contents

1 **DTV Design Library**
   - Introduction............................................................................................................... 1-1
   - ISDB-T System ......................................................................................................... 1-1
   - DVB-T System .......................................................................................................... 1-2
   - OFDM Technique ...................................................................................................... 1-3
   - Glossary of Terms .................................................................................................... 1-5
   - References ............................................................................................................... 1-6

2 **Channel Coding Components**
   - DTV_BCHCoder ....................................................................................................... 2-2
   - DTV_BCHDecoder ................................................................................................... 2-5
   - DTV_ConvCoder ...................................................................................................... 2-10
   - DTV_ConvCoder1_2 ............................................................................................... 2-12
   - DTV_ConvDecoder .................................................................................................. 2-15
   - DTV_ConvDecoder1_2 ............................................................................................ 2-18
   - DTV_Demapper........................................................................................................ 2-20
   - DTV_DQPSKCoder .................................................................................................. 2-23
   - DTV_DQPSKDecoder .............................................................................................. 2-25
   - DTV_InnerDecoder................................................................................................... 2-27
   - DTV_InterlvFloat....................................................................................................... 2-29
   - DTV_InterlvInt ......................................................................................................... 2-31
   - DTV_Mapper ............................................................................................................ 2-34
   - DTV_PNreset ........................................................................................................... 2-41
   - DTV_PuncCoder ...................................................................................................... 2-43
   - DTV_PuncConvCoder .............................................................................................. 2-45
   - DTV_PuncConvDecoder .......................................................................................... 2-47
   - DTV_PuncDecoder ................................................................................................... 2-50
   - DTV_QAM16Coder .................................................................................................. 2-52
   - DTV_QAM16Decoder ............................................................................................... 2-56
   - DTV_QAM64Coder .................................................................................................. 2-58
   - DTV_QAM64Decoder ............................................................................................... 2-62
   - DTV_QPSKCoder ..................................................................................................... 2-64
   - DTV_QPSKDecoder ................................................................................................ 2-66
   - DTV_RSCoder .......................................................................................................... 2-67
   - DTV_RSDecoder ..................................................................................................... 2-71
   - DTV_ScrambleByte .................................................................................................. 2-78

3 **DVB-T Components**
   - DTV_CIRNorm ......................................................................................................... 3-2
   - DTV_DVBBitBlockInterlv ....................................................................................... 3-4
4 ISDB-T Components
DTV_CDSCDecoder ......................................................... 4-2
DTV_CarrierRotator .......................................................... 4-7
DTV_CarrierScrambler ........................................................ 4-9
DTV_ChEstimator ............................................................. 4-14
DTV_DemuxCohSegs ................................................................ 4-20
DTV_DemuxDiffSegs ........................................................... 4-26
5 Multiplex Components
DTV_CommCtrl2 ................................................................. 5-2
DTV_CommCtrl3 ................................................................. 5-3
DTV_DistCtrl2 ................................................................. 5-4
DTV_DistCtrl3 ................................................................. 5-5
DTV_SplitThreeLayData .................................................. 5-6
DTV_SplitThreeLayTSP .................................................... 5-8
DTV_SplitTwoLayData ....................................................... 5-11
DTV_SplitTwoLayTSP ....................................................... 5-13
DTV_SynLayTMCC1 .......................................................... 5-16
DTV_SynLayTMCC2 .......................................................... 5-18
DTV_SynLayTMCC3 .......................................................... 5-20
DTV_SynThreeLayData ..................................................... 5-22
DTV_SynThreeLayTSP ..................................................... 5-24
DTV_SynTwoLayData ....................................................... 5-27
DTV_SynTwoLayTSP ....................................................... 5-29

6 OFDM Components
DTV_AddFixPhase ............................................................ 6-2
DTV_InsertGuard .............................................................. 6-4
DTV_LoadIFFTBuff ............................................................ 6-6
DTV_MLEstimator ............................................................. 6-10
DTV_OFDMEqualizer ......................................................... 6-13
DTV_RemovePhase ............................................................ 6-15

7 Test Components
DTV_BER ......................................................................... 7-2
DTV_PowerMeasure .......................................................... 7-4

8 DVB-T Design Examples
Introduction ..................................................................... 8-1
DTV_OFDM ....................................................................... 8-2
OFDM 16-QAM Modulation and Demodulation in DVB-T Systems .................. 8-3
OFDM 64-QAM Modulation and Demodulation in DVB-T Systems ............... 8-9
OFDM, 64-QAM DTV and PAL Signal Interference Test in DTV Systems .... 8-16
DTV DVB System ....................................................................................... 8-21
OFDM 16-QAM DVB-T System without Channel Coding BER ...................... 8-22
16-QAM DVB-T System Design ................................................................ 8-30
Hierarchical 64-QAM DVB-T System Design ........................................... 8-42

9 ISDB-T Design Examples
Introduction ............................................................................................... 9-1
DTV ISDB OFDM ....................................................................................... 9-2
OFDM 64-QAM Modulation and Demodulation in ISDB-T Systems ............ 9-8
OFDM DQPSK Modulation and Demodulation in ISDB-T Systems .............. 9-14
DTV and NTSC Signal Interference Test in ISDB-T Systems ..................... 9-20
OFDM 2-Layer Modulation and Demodulation in ISDB-T Systems .......... 9-25
OFDM 3-Layer Modulation and Demodulation in ISDB-T Systems .......... 9-31
DTV ISDB System ....................................................................................... 9-31
OFDM 64-QAM ISDB-T System without Channel Coding BER .................. 9-36
OFDM DQPSK ISDB-T System without Channel Coding BER ................... 9-41
1-Layer 64-QAM Mapping ISDB-T System Design .................................. 9-46
1-Layer DQPSK-Mapping ISDB-T System Design .................................... 9-52
3-Layer ISDB-T System Design ................................................................ 9-57
2-Layer ISDB-T System Design ................................................................ 9-70
TMCC in ISDB-T 1-Layer System Design ................................................ 9-75
TMCC in ISDB-T 3-Layer System Design ................................................ 9-83

10 DVB-T Receiver Design Examples
DVB-T AWGN Performance ....................................................................... 10-2
DVB-T Digital Adjacent Channel Performance ......................................... 10-6
DVB-T Impulse Interference Performance .................................................. 10-13
DVB-T Long Delay Channel Performance ................................................ 10-18
DVB-T Analog Adjacent Channel Performance ........................................ 10-22
DVB-T Analog Co-Channel Performance .................................................. 10-28
DVB-T Image Channel Performance .......................................................... 10-35
DVB-T Rayleigh Channel (P1) Performance ............................................... 10-39
DVB-T Ricean Channel (F1) Performance .................................................. 10-44
DVB-T Receiver Minimum Input Level Sensitivity .................................... 10-48
DVB-T Short Delay Channel Performance ............................................... 10-53
DVB-T Single Echo Channel Performance ............................................... 10-57
Chapter 1: DTV Design Library

Introduction

The Agilent EEsof DTV Design Library includes the current Japanese, and European HDTV standards for the Advanced Design System platform. This design library focuses on the transmission layer of the HDTV system and is intended to be a baseline system for designers to get an idea of what a nominal or ideal system performance would be. They also provide determination of degraded system performance due to system impairments that may include nonideal component performance.

ISDB-T System

The ISDB-T system has been designed to have the flexibility to send television or sound programs as digital signals while offering multimedia services in which a variety of digital information (video, sound, text and computer programs) can be integrated. It aims to make use of the advantages provided by terrestrial radio waves so that stable reception can be provided by compact, light and inexpensive mobile receivers in addition to integrated receivers used at home by using a BST (band segmented transmission)-OFDM scheme [1].

Two transmission bandwidths are prescribed (5.6 MHz and 432 kHz), each oriented to particular types of broadcasting services. The 5.6-MHz bandwidth is primarily for digital broadcasting of television programs; the 432-kHz bandwidth is primarily for audio programs. These two modes share all other parameters such as encoding format, multiplexing format, and OFDM carrier interval and frame configuration.

Terrestrial ISDB provides hierarchical transmission features using different carrier modulation schemes (DQPSK, QPSK, 16-QAM, 64-QAM) and internal encoding rates (1/2, 2/3, 3/4, 5/6, 7/8). This enables part of the band to be allocated to signals for stationary reception and the rest to signals for mobile reception; this means audio and data broadcasts for automobile and portable receivers can be performed simultaneously with television broadcasts for home use. Each hierarchical level can be set for each BST segment having a bandwidth of 432 kHz. Such information can be sent to receivers by TMCC (transmission and multiplexing configuration control) signal allocated to part of the OFDM carrier.
Because the wide and narrow bandwidths in terrestrial ISDB share the same OFDM parameters, the 5.6-MHz wide band can directly include the 432-kHz narrow band. Consequently, a 432-kHz receiver can receive some 5.6-MHz services, and a 5.6-MHz receiver can receive all services broadcast at 432 kHz.

Based on the system configuration, this design library may include three sub-libraries, transmission, receiving and channel coding sub-libraries.

**DVB-T System**

The DVB-T system was designed with the flexibility to adapt to all channels including clear channels and interleaved planning, and co-channel operation for the same program by different transmitters (single-frequency networks).

It also permits service flexibility, with the possibility of reception by roof-top antennae and portable reception, if necessary. Mobile reception is possible for QPSK and for higher modulation orders, as proven by extensive laboratory measurements and field trials under different channel conditions.

The system was also designed to be robust against interference from delayed signals, either echoes from terrain or buildings or signals from distant transmitters in a single frequency network, a new tool which it brings to TV service planning to improve spectrum efficiency which is necessary in the case of particularly crowded spectrum as it is the case in Europe.

The DVB-T compliant signals can also be carried over cables. However, the DVB-T specification is part of a family of specifications covering also satellite (DVB-S) and cable (DVB-C) operation. All use MPEG-2 coding for video and audio and MPEG-2 type of multiplexing. They have common features in the error protection strategy to be used. The main difference is the modulation method that is specific to the relevant bearer (satellite, cable, or terrestrial). The available data capacity is also different, as higher bit rates are offered on cable and satellite. However, transferring programs from one bearer to another is possible provided that the bit rate is available.

The DVB-T system features a number of selectable parameters so it can accommodate a large range of carrier-to-noise ratio and channel behavior, enabling fixed, portable, or mobile reception, with a trade-off in the usable bit rate. The range of parameters lets broadcasters select a mode appropriate to the application. For example, a very robust mode (with correspondingly lower payload) is needed to ensure portable reception. A moderately robust mode with a higher payload could be used where the service planning uses interleaved channels. The less robust modes
with the highest payloads can be used if a clear channel is available for digital TV broadcasting.

**OFDM Technique**

Multi-carrier, or orthogonal frequency-division multiplexing (OFDM), systems have gained in interest during the last years. It is used in the European digital broadcast radio system, and its use in wireless applications such as digital broadcast television and mobile communication systems is currently being investigated. By the name of discrete multi-tone (DMT) modulation, OFDM is also being examined for broadband digital communication on existing copper networks. The OFDM technique has been proposed for high bit-rate digital subscriber lines (HDSL) and for asymmetric digital subscriber lines (ADSL).

The OFDM concept is based on spreading the data to be transmitted over a large number of carriers, each being modulated at a low bit rate. The multiplex of carriers can be conveniently generated digitally using the inverse fast-Fourier-transform (FFT) process.

Preferred implementations of the FFT tend to be based on radix 2 or radix 4 algorithms, or some combination of radix 2 and 4. Example systems are based on 2048 (2k) carriers and 8192 (8k) carriers. However, the number of actual carriers transmitted is always smaller than the maximum number possible, as some carriers at either edge of the channel are not used. These unused carriers make a frequency guard band for practical IF filtering. The active carriers carry data or synchronization information. Any digital modulation scheme can be used to modulate the active carriers, for example, QPSK, n-QAM, where n is commonly 16 or 64.

OFDM, due to its multi-carrier nature, exhibits relatively long symbol periods. This long symbol period provides a degree of protection against inter-symbol interference caused by multi-path propagation. However, this protection can be greatly enhanced by use of the guard interval. The guard interval is a cyclic extension of the symbol, in simplistic terms a section of the start of the symbol is simply added to the end of the symbol.

OFDM, when coupled with appropriate channel coding (error correction coding), can achieve a high level of immunity against multi-path propagation and against co-channel interference, for example, NTSC, PAL, SECAM. OFDM systems also offer the broadcaster great flexibility as bit rates can be traded against level of protection depending on the nature of the service. For example, mobile reception of the OFDM signal may be possible given due consideration to factors including vehicle speed,
carrier spacing, data rate and modulation scheme; whereas, for a service with fixed reception, high order modulation schemes and consequently high data rates could be used.

OFDM signals also allow the possibility of single-frequency network (SFN) operation. This is due to OFDM multi-path immunity. SFN operation is possible when exactly the same signal, in time and frequency, is radiated from multiple transmitters. In this case at any reception point in the coverage overlap between transmitters, the weakest received signals will act as post- or pre-echoes to the strongest signal. However, if the transmitters are far apart the time delay between the received signals will be large and the system will need a large guard interval.

While the use of the guard interval (or cyclic prefix) removes the effects of inter-symbol interference under multi-path conditions, it cannot remove the effects of frequency selective fading. Under these conditions the amplitude and phase of each subcarrier is distorted. If the OFDM receiver is to coherently demodulate the signal it must equalize the phase and amplitude of each carrier; this can be done after the FFT using a simple equalizer. This process is known as channel estimation and equalization. The block diagram of a basic OFDM system is shown in Figure 1-1.
## Glossary of Terms

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AC</td>
<td>Auxiliary Channel</td>
</tr>
<tr>
<td>ACI</td>
<td>Adjacent Channel Interference</td>
</tr>
<tr>
<td>ACPR</td>
<td>Adjacent Channel Power Ratio</td>
</tr>
<tr>
<td>AFC</td>
<td>Automatic Frequency Control</td>
</tr>
<tr>
<td>ARIB</td>
<td>Association of Radio Industries and Business</td>
</tr>
<tr>
<td>AWGN</td>
<td>Additive White Gaussian Noise</td>
</tr>
<tr>
<td>BCH</td>
<td>Bose-Chaudhuri-Hocquenghem code</td>
</tr>
<tr>
<td>BER</td>
<td>Bit Error Rate</td>
</tr>
<tr>
<td>CP</td>
<td>Continual Pilot</td>
</tr>
<tr>
<td>CDSC</td>
<td>Complete Differential Set Code</td>
</tr>
<tr>
<td>DBPSK</td>
<td>Differential Binary Phase Shift Keying</td>
</tr>
<tr>
<td>DQPSK</td>
<td>Differential Quadrature Phase Shift Keying</td>
</tr>
<tr>
<td>DFT</td>
<td>Discrete Fourier Transform</td>
</tr>
<tr>
<td>DVB</td>
<td>Digital Video Broadcasting</td>
</tr>
<tr>
<td>DVB-T</td>
<td>DVB-Terrestrial</td>
</tr>
<tr>
<td>EDTV</td>
<td>Enhanced Definition TeleVision</td>
</tr>
<tr>
<td>ETS</td>
<td>European Telecommunication Standard</td>
</tr>
<tr>
<td>EVM</td>
<td>Error Vector Magnitude</td>
</tr>
<tr>
<td>FEC</td>
<td>Forward Error Correction</td>
</tr>
<tr>
<td>FFT</td>
<td>Fast Fourier Transform</td>
</tr>
<tr>
<td>FIFO</td>
<td>First-In, First-Out shift register</td>
</tr>
<tr>
<td>HDTV</td>
<td>High Definition TeleVision</td>
</tr>
<tr>
<td>HP</td>
<td>High Priority bit stream</td>
</tr>
<tr>
<td>IFFT</td>
<td>Inverse Fast Fourier Transform</td>
</tr>
<tr>
<td>ISDB-T</td>
<td>Terrestrial Integrated Services Digital Broadcasting</td>
</tr>
<tr>
<td>LP</td>
<td>Low Priority bit stream</td>
</tr>
<tr>
<td>MPEG</td>
<td>Moving Picture Experts Group</td>
</tr>
<tr>
<td>NTSC</td>
<td>National Television System Committee</td>
</tr>
<tr>
<td>OFDM</td>
<td>Orthogonal Frequency Division Multiplexing</td>
</tr>
<tr>
<td>PAL</td>
<td>Phase Alternating Line</td>
</tr>
<tr>
<td>QAM</td>
<td>Quadrature Amplitude Modulation</td>
</tr>
<tr>
<td>QPSK</td>
<td>Quadrature Phase Shift Keying</td>
</tr>
<tr>
<td>RS</td>
<td>Reed-Solomon</td>
</tr>
<tr>
<td>SFN</td>
<td>Single Frequency Network</td>
</tr>
</tbody>
</table>
DTV Design Library

<table>
<thead>
<tr>
<th>SP</th>
<th>Scattered Pilot</th>
</tr>
</thead>
<tbody>
<tr>
<td>TSP</td>
<td>Transport Stream Packet</td>
</tr>
<tr>
<td>TS</td>
<td>Transport Stream</td>
</tr>
<tr>
<td>TMCC</td>
<td>Transmission and Multiplexing Configuration Control</td>
</tr>
<tr>
<td>TPS</td>
<td>Transmission Parameter Signalling</td>
</tr>
<tr>
<td>UHF</td>
<td>Ultra-High Frequency</td>
</tr>
<tr>
<td>VHF</td>
<td>Very-High Frequency</td>
</tr>
</tbody>
</table>

References


Chapter 2: Channel Coding Components
Channel Coding Components

DTV_BCHCoder

Description  BCH coder
Library       DTV, Channel Coding
Class         SDFDTV_BCHCoder

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Default</th>
<th>Sym</th>
<th>Type</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>FieldOrder</td>
<td>order of Galois field</td>
<td>7</td>
<td>M</td>
<td>int</td>
<td>[3, 12]</td>
</tr>
<tr>
<td>InfoLength</td>
<td>information bit data length</td>
<td>113</td>
<td>K</td>
<td>int</td>
<td>[1, N†]</td>
</tr>
<tr>
<td>BlockLength</td>
<td>length of block code</td>
<td>127</td>
<td>N</td>
<td>int</td>
<td>([2^{*(M-1)}, 2^M))</td>
</tr>
<tr>
<td>ErrorNum</td>
<td>error protection capability</td>
<td>2</td>
<td></td>
<td>int</td>
<td>([0, 7])</td>
</tr>
</tbody>
</table>

† InfoLength value depends on ErrorNum and BlockLength.

Pin Inputs

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>input</td>
<td>input signal</td>
<td>int</td>
</tr>
</tbody>
</table>

Pin Outputs

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>output</td>
<td>error protected output signal</td>
<td>int</td>
</tr>
</tbody>
</table>

Notes/Equations

1. This model is used to perform shortened BCH error correcting encoding over the input signal. Each firing, K tokens are consumed at the input pin and N tokens are produced; N–K parity bits are generated and appended to the input signal to form the output.

2. Implementation
Add 0 bit to make the output signal block length up to $2^M - 1$.

If the output signal block length is different with $(2**M - 1)$, the input signal block is added $(2**M - 1 - N)$ bits which are set to zero, and consists of a new BCH code length $2^M - 1$.

Calculate the output BCH code by field generator polynomial $g(x)$.

Calculate the coefficients of redundancy polynomial $b(x)$. The redundancy polynomial $b(x)$ is the remainder after dividing $(x^{N-K} \times \text{data}(x))$ by the generator polynomial $g(x)$.

The output BCH $(2^M - 1, 2^M - 1 - N + K)$ code polynomial is

$$A(x) = x^{N-K} \times \text{data}(x) + b(x)$$

where

- $\text{data}(x)$ denotes information bits data polynomial
- $A(x)$ denotes the output data polynomial
- $b(x)$ denotes redundancy polynomial

The added 0 bits are deleted from the BCH $(2^M - 1, 2^M - 1 - N + K)$ and the output shortened BCH $(N,K)$ code is determined.

3. DVB-T Primitive and Generation Polynomials

For DVB-T, the shortened BCH $(67,53)$ is derived from BCH $(127,113)$. The shortened BCH code is implemented by adding 60 bits, all set to zero, before the information bits input of a BCH $(127,113, t = 2)$ encoder. After BCH encoding, these null bits are discarded, leading to a BCH code word of 67 bits.

For the primitive polynomial

$$P(x) = x^7 + x^3 + 1$$

For the generation polynomial

$$g(x) = x^{14} + x^9 + x^8 + x^6 + x^5 + x^4 + x^2 + x + 1$$

The primitive polynomials of BCH codes based on FieldOrder values are given in Table 2-1.
## Channel Coding Components

### References


### Table 2-1. Primitive Polynomial of BCH Coding

<table>
<thead>
<tr>
<th>FieldOrder</th>
<th>Primitive Polynomial</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>( P(x) = x^3 + x + 1 )</td>
</tr>
<tr>
<td>4</td>
<td>( P(x) = x^4 + x + 1 )</td>
</tr>
<tr>
<td>5</td>
<td>( P(x) = x^5 + x^2 + 1 )</td>
</tr>
<tr>
<td>6</td>
<td>( P(x) = x^6 + x + 1 )</td>
</tr>
<tr>
<td>7</td>
<td>( P(x) = x^7 + x^3 + 1 )</td>
</tr>
<tr>
<td>8</td>
<td>( P(x) = x^8 + x^6 + x^5 + x^4 + 1 )</td>
</tr>
<tr>
<td>9</td>
<td>( P(x) = x^9 + x^4 + 1 )</td>
</tr>
<tr>
<td>10</td>
<td>( P(x) = x^{10} + x^3 + 1 )</td>
</tr>
<tr>
<td>11</td>
<td>( P(x) = x^{11} + x^2 + 1 )</td>
</tr>
<tr>
<td>12</td>
<td>( P(x) = x^{12} + x^7 + x^4 + x^3 + 1 )</td>
</tr>
</tbody>
</table>
DTV_BCHDecoder

Description  BCH decoder
Library       DTV, Channel Coding
Class         SDFDTV_BCHDecoder

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Default</th>
<th>Sym</th>
<th>Type</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>FieldOrder</td>
<td>order of Galois field</td>
<td>7</td>
<td>M</td>
<td>int</td>
<td>[3, 12]</td>
</tr>
<tr>
<td>InfoLength</td>
<td>information bit data length</td>
<td>113</td>
<td>K</td>
<td>int</td>
<td>[1, N]†</td>
</tr>
<tr>
<td>BlockLength</td>
<td>length of block code</td>
<td>127</td>
<td>N</td>
<td>int</td>
<td>(2^{(M-1)}, 2^M)</td>
</tr>
<tr>
<td>ErrorNum</td>
<td>error protection capability</td>
<td>2</td>
<td>N</td>
<td>int</td>
<td>[0, 7]</td>
</tr>
</tbody>
</table>

† InfoLength value depends on ErrorNum and BlockLength.

Pin Inputs

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>input</td>
<td>input signal</td>
<td>int</td>
</tr>
</tbody>
</table>

Pin Outputs

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>output</td>
<td>decoded signal</td>
<td>int</td>
</tr>
</tbody>
</table>

Notes/Equations

1. This model is used to perform BCH error correcting decoding over the input signal. After decoding, N tokens are consumed at the input pin and K tokens are produced each firing.

2. Implementation
When input data is shortened BCH code at the start of decoding, the input signal block is added \((2^M-1-N)\) bits and consists of BCH \((2^M-1, 2^M-1-N+K)\) code. After decoding, the padded bits are discarded.

**Syndrome Concept**

How can a receiver know the occurrence of one or more errors during transmission/storage of an encoded block, given only the generator polynomial \(g(D)\) and the received code word \(y\) represented by its polynomial \((D)\)?

The answer is to become familiar with syndromes, which indicate an erroneous situation. Suppose code word \(x(D)\) is transmitted and we receive \(y(D)\). We can write \(y(D) = X(D) + e(D)\) where \(e(D)\) represents the polynomial corresponding to the error that occurred during transmission.

Now we have a simple procedure for checking the occurrence of errors at the receiver as follows.

- Transform the received code word \(y\) into its polynomial representation \(y(D)\);
- Calculate \(S(i) = y(\alpha^i) = e(\alpha^i)\) for \(i = 1, \ldots, 2t\);

if one or more \(S(i)\) values are not equal to zero, one or more symbol errors have occurred in the block.

Now we know if there are errors in the received block, but we don't know how many bits are affected, or the position of the errors. The location of symbol errors in a block is determined by the construction of another polynomial, the error locator \(A(D)\).

Given the \(2t\) syndromes \(S(i)\), the decoding algorithm will try to synthesize a polynomial of which the coefficient values indicate the error positions. This is done as follows. Let \(n\) errors occur in locations \(j_1, j_2, \ldots, j_n\) of a block, where \(0 \leq j_0 < j_1 < \ldots < j_n < N\). Then the error polynomial \(e(D)\) is:

\[ e(D) = D^{j_0} + D^{j_0-1} + \ldots + D^{j_1} \]

Define the error locator \(X(l)\) by \(X_l = \alpha^{j_l} \) for \(l = 1, 2, \ldots, n\)

Then the syndromes can be expressed in terms of these error locators:

\[ S(i) = e(\alpha^i) = X_n^i + X_{n-1}^i + \ldots + X_1^i \quad i = 1, \ldots, 2t \]

We now construct the error locator polynomial \(A(D)\) as follows:
This polynomial is the error locator polynomial since the inverses of its roots, \( X(l) \) \((l=1, \ldots, n)\), yield the error locations.

To construct \( A(D) \), this model uses the Massey-Berlekamp algorithm, which forms the key to decoding BCH code. The algorithm is more generally applicable for synthesizing linear shift feedback registers generating a predefined output sequence.

For the Massey-Berlekamp Algorithm an iterative table will be filled as shown in Table 2-2.

<table>
<thead>
<tr>
<th>( \mu )</th>
<th>( A^{(\mu)}(D) )</th>
<th>( d_\mu )</th>
<th>( l_\mu )</th>
<th>( \mu-l_\mu )</th>
</tr>
</thead>
<tbody>
<tr>
<td>-1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>-1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>\vdots</td>
<td>\vdots</td>
<td>\vdots</td>
<td>\vdots</td>
<td>\vdots</td>
</tr>
<tr>
<td>2t</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

where \( \mu \) is the iterative step number, \( d_\mu \) is the \( \mu \)-th step iterative difference, \( l_\mu \) is the order of \( A^{(\mu)}(D) \).

- If \( d_\mu = 0 \)
  
  then \( A^{(\mu+1)}(D) = A^{(\mu)}(D) \) and \( l_{\mu+1} = l_\mu \).

- If \( d_\mu \neq 0 \), search for lines in the table to find step \( \rho \) in which \( d_\rho \neq 0 \) and the value of \( \rho - l_\rho \) is the maximum, then
  
  \[
  A^{(\mu+1)}(D) = A(D) - d_\mu d_\rho^{-1} D^{(\mu - \rho)} \Omega_1(D)
  \]
Channel Coding Components

and

\[ l_{\mu + 1} = \max(l_\mu, l_\rho + \mu - \rho) \]

For the two conditions

\[ d_{\mu + 1} = s_{\mu + 2} + A^{(\mu + 1)}_{l_{\mu + 1}} s_{\mu + 1} + \ldots + A^{(\mu + 1)}_{l_{\mu}} s_{\mu + 2 - l_{\mu + 1}} \]

Iterate until the last line of the table \( A^{(2l)} \) (D) is determined. If the order of the polynomial is greater than \( t \), which means the received code word block has more than \( t \) errors, the errors cannot be corrected.

To decode binary BCH code, it is sufficient to know the position of bit errors in a block and make the bit value inverse.

3. DVB-T Primitive and Generation Polynomials

For DVB-T, the shortened BCH(67,53) is derived from BCH(127,113). The shortened BCH code is implemented by adding 60 bits, all set to zero, before the information bits input of a BCH(127,113, \( t = 2 \)) encoder. After BCH encoding, these null bits are discarded, leading to a BCH code word of 67 bits.

For the primitive polynomial

\[ P(x) = x^7 + x^3 + 1 \]

For the generation polynomial

\[ g(x) = x^{14} + x^9 + x^8 + x^6 + x^5 + x^4 + x^2 + x + 1 \]

The primitive polynomials of BCH codes based on FieldOrder values are listed in Table 2-3.

<table>
<thead>
<tr>
<th>FieldOrder</th>
<th>Primitive Polynomial</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>( P(x) = x^2 + x + 1 )</td>
</tr>
<tr>
<td>3</td>
<td>( P(x) = x^3 + x + 1 )</td>
</tr>
<tr>
<td>4</td>
<td>( P(x) = x^4 + x + 1 )</td>
</tr>
<tr>
<td>5</td>
<td>( P(x) = x^5 + x^2 + 1 )</td>
</tr>
<tr>
<td>6</td>
<td>( P(x) = x^6 + x + 1 )</td>
</tr>
<tr>
<td>7</td>
<td>( P(x) = x^7 + x^3 + 1 )</td>
</tr>
</tbody>
</table>
Table 2-3. Primitive Polynomials of BCH Coding

<table>
<thead>
<tr>
<th>FieldOrder</th>
<th>Primitive Polynomial</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>( p(x) = x^8 + x^6 + x^5 + x^4 + 1 )</td>
</tr>
<tr>
<td>9</td>
<td>( p(x) = x^9 + x^4 + 1 )</td>
</tr>
<tr>
<td>10</td>
<td>( p(x) = x^{10} + x^3 + 1 )</td>
</tr>
<tr>
<td>11</td>
<td>( p(x) = x^{11} + x^2 + 1 )</td>
</tr>
<tr>
<td>12</td>
<td>( p(x) = x^{12} + x^7 + x^4 + x^3 + 1 )</td>
</tr>
<tr>
<td>13</td>
<td>( p(x) = x^{13} + x^4 + x^3 + x + 1 )</td>
</tr>
<tr>
<td>14</td>
<td>( p(x) = x^{14} + x^{12} + x^{11} + x + 1 )</td>
</tr>
<tr>
<td>15</td>
<td>( p(x) = x^{15} + x + 1 )</td>
</tr>
<tr>
<td>16</td>
<td>( p(x) = x^{16} + x^5 + x^3 + x^2 + 1 )</td>
</tr>
<tr>
<td>17</td>
<td>( p(x) = x^{17} + x^3 + x )</td>
</tr>
<tr>
<td>18</td>
<td>( p(x) = x^{18} + x^7 + x )</td>
</tr>
<tr>
<td>19</td>
<td>( p(x) = x^{19} + x^6 + x^5 + x + 1 )</td>
</tr>
<tr>
<td>20</td>
<td>( p(x) = x^{20} + x^3 + 1 )</td>
</tr>
</tbody>
</table>

References

Channel Coding Components

**DTV_ConvCoder**

Description: Convolutional coding the input bits
Library: DTV, Channel Coding
Class: SDFDTV_ConvCoder
Required Licenses

**Pin Inputs**

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>input</td>
<td>bits to be coded</td>
<td>int</td>
</tr>
</tbody>
</table>

**Pin Outputs**

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>output</td>
<td>coded bits</td>
<td>int</td>
</tr>
</tbody>
</table>

**Notes/Equations**

1. This model is used to perform normal convolutional encoding of data rate 1/2 over the input signal.
   Each firing, 1 token is consumed at the input and 2 tokens are produced.
   The encoder is illustrated in Figure 2-1.
Figure 2-1. Convolutional Code Rate 1/2
(Constraint Length=7)

References

Channel Coding Components

DTV_ConvCoder1_2

Description  DTV convolutional encoder for 1/2 rate
Library     DTV, Channel Coding

Pin Inputs

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>input</td>
<td>signal to be encoded</td>
<td>int</td>
</tr>
</tbody>
</table>

Pin Outputs

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>output</td>
<td>encoded signal</td>
<td>int</td>
</tr>
</tbody>
</table>

Notes/Equations

1. This subnetwork model is used to perform normal convolutional encoding of data rate 1/2 over the input signal. Each firing, 1 token is consumed at the input and 2 tokens are produced. The schematic for this subnetwork is shown in Figure 2-2.
This subnetwork uses a general convolutional coding model to encode the input data into mother convolutional code of data rate 1/2. The encoder is illustrated in Figure 2-3.

References

Channel Coding Components

**DTV_ConvDecoder**

**Description**  Bit by bit viterbi decoder for DTV convolutional code

**Library**  DTV, Channel Coding

**Class**  SDF DTV_ConvDecoder

**Derived From**  DTV_ConvDecoder_base

**Required Licenses**

**Parameters**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Default</th>
<th>Type</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>TrunLen</td>
<td>path memory truncation length (bytes)</td>
<td>10</td>
<td>int</td>
<td>[5, ∞)</td>
</tr>
</tbody>
</table>

**Pin Inputs**

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>input</td>
<td>The code words to be viterbi-decoded.</td>
<td>real</td>
</tr>
</tbody>
</table>

**Pin Outputs**

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>output</td>
<td>the decoded bits.</td>
<td>int</td>
</tr>
</tbody>
</table>

**Notes/Equations**

1. This component is used to Viterbi-decode the input code words. CC(2,1,7) and g1 171 g2 133 is decoded. The delay is equal to the convolutional code memory. Padding bits detect the end of the code words.

One output token is produced when 2 input tokens are consumed.

2. The following algorithm is used, with CC(2,1,7) as an example. The generator functions of the code are g1 (which equals 171 (octal)), and g2 (which equals 133 (octal)).
Channel Coding Components

Because the constraint length is 7, there are 64 possible states in the encoder. In the Viterbi decoder all states are represented by a single column of nodes in the trellis at each symbol instant. At each node in the trellis, there are 2 merging paths; the path with the shortest distance is selected as the survivor.

The long encoded packets in DTV systems make it impractical to store the entire length of the surviving sequences before determining the information sequence when decoding delay and memory is concerned. Instead, only the most recent L information bits in each surviving sequence are stored. Once the path with the shortest distance is identified, the symbol associated with path L periods ago is conveyed to the output as a decoded information symbol. Generally, parameter L is sufficiently large (normally $L \geq 5K$) for the present symbol of the surviving sequences to have a minimum effect on the decoding of the $L$th previous symbol. In DTV systems, $L = 8 \times $TrunLen.

The following is the Viterbi algorithm for decoding a CC($n,k,K$) code, where $K$ is the constraint length of convolutional code. In our components, the convolutional code is processed with $k=1$.

**Branch Metric Calculation**

Branch metric $m^{(\alpha)}_j$, at the $J$th instant of the $\alpha$ path through the trellis is defined as the logarithm of the joint probability of the received $n$-bit symbol $r_{j1}, r_{j2}, \ldots, r_{jn}$ conditioned on the estimated transmitted $n$-bit symbol $c_{j1}^{(\alpha)}, c_{j2}^{(\alpha)}, \ldots, c_{jn}^{(\alpha)}$ for the $\alpha$ path. That is,

$$m^{(\alpha)}_j = \ln \left( \prod_{i=1}^{n} P(r_{ji}|c_{ji}^{(\alpha)}) \right) = \sum_{i=1}^{n} \ln P(r_{ji}|c_{ji}^{(\alpha)}).$$

If the receiver is regarded as part of the channel, for the Viterbi decoder the channel can be considered as an AWGN channel. Therefore,

$$m^{(\alpha)}_j = \sum_{i=1}^{n} r_{ji} c_{ji}$$
Path Metric Calculation

The path metric $M(\alpha)$ for the $\alpha$ path at the $J$th instant is the sum of the branch metrics belonging to the $\alpha$ path from the first to the $J$th instant. Therefore,

$$M(\alpha) = \sum_{j=1}^{J} m(\alpha)_j$$

Information Sequence Update

There are $2^k$ merging paths at each node in the trellis and the decoder selects the one (known as the survivor) with the largest metric from paths $\alpha_1, \alpha_2, ..., \alpha_{2^k}$; namely

$$\max(M(\alpha_1), M(\alpha_2), ..., M(\alpha_{2^k}))$$

Decoder Output

When the two survivors are determined at the $J$th instant, the decoder outputs the $(J-L)$th information symbol from its memory of the survivor with the largest metric.

References


Channel Coding Components

**DTV_ConvDecoder1_2**

**Description**  DTV convolutional decoder for 1/2 rate  
**Library**  DTV, Channel Coding

**Parameters**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Default</th>
<th>Type</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>NumBits</td>
<td>number of soft decision bits</td>
<td>4</td>
<td>int</td>
<td>[1, ∞)</td>
</tr>
<tr>
<td>SymbolLen</td>
<td>path memory truncation length</td>
<td>8</td>
<td>int</td>
<td>[1, ∞)</td>
</tr>
</tbody>
</table>

**Pin Inputs**

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>input</td>
<td>signal to be decoded</td>
<td>int</td>
</tr>
</tbody>
</table>

**Pin Outputs**

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>output</td>
<td>decoded signal</td>
<td>int</td>
</tr>
</tbody>
</table>

**Notes/Equations**

1. This subnetwork model is used to perform Viterbi decoding of convolutional code that has a mother convolutional code of data rate 1/2 over the input signal. The schematic for this subnetwork is shown in Figure 2-4.
2. A general Viterbi convolutional decoding model decodes the convolutional encoded input data. The encoder is illustrated in Figure 2-5.

Figure 2-5. Mother Convolutional Code of Rate 1/2 (Constraint Length=7)

References


Channel Coding Components

**DTV_Demapper**

**Description**  Soft demapper for uniform and non-uniform QPSK, 16QAM and 64QAM

**Library**  DTV, Channel Coding

**Class**  SDFDTV_Demapper

**Required Licenses**

**Parameters**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Default</th>
<th>Type</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>MappingMode</td>
<td>signal constellations and mapping: QPSK, QAM-16, QAM-64</td>
<td>QAM-16</td>
<td>enum</td>
<td></td>
</tr>
<tr>
<td>Alpha</td>
<td>non-uniform factor for DVB-T</td>
<td>1</td>
<td>int</td>
<td>[1, ∞)</td>
</tr>
</tbody>
</table>

**Pin Inputs**

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>input</td>
<td>signal to be demodulated</td>
<td>complex</td>
</tr>
<tr>
<td>2</td>
<td>CSI</td>
<td>channel state information</td>
<td>real</td>
</tr>
</tbody>
</table>

**Pin Outputs**

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>output</td>
<td>soft metric information</td>
<td>real</td>
</tr>
</tbody>
</table>

**Notes/Equations**

1. This model is used to de-map uniform and non-uniform QPSK, 16QAM and 64QAM to determine the soft bit information that is used by Viterbi decoding.

   Each firing, the CSI and input pins consume 1 token each; 2 tokens for QPSK, 4 tokens for 16QAM, or 6 tokens for 64QAM are generated.

2. MappingMode specifies the signal constellation and mapping mode: QPSK, QAM-16 or QAM-64.
While the constellation point is sent to the input port before demapping, the point is scaled up by a factor that is determined by Alpha and MappingMode given in Table 2-4.

Table 2-4. De-normalization Factors for Data Symbols

<table>
<thead>
<tr>
<th>Modulation scheme</th>
<th>Normalization factor</th>
</tr>
</thead>
<tbody>
<tr>
<td>QPSK</td>
<td>$c = z \times (\sqrt{2})$</td>
</tr>
<tr>
<td>16-QAM: Alpha=1</td>
<td>$c = z \times (\sqrt{10})$</td>
</tr>
<tr>
<td>16-QAM: Alpha=2</td>
<td>$c = z \times (\sqrt{50})$</td>
</tr>
<tr>
<td>16-QAM: Alpha=4</td>
<td>$c = z \times (\sqrt{52})$</td>
</tr>
<tr>
<td>64-QAM: Alpha=1</td>
<td>$c = z \times (\sqrt{42})$</td>
</tr>
<tr>
<td>64-QAM: Alpha=2</td>
<td>$c = z \times (\sqrt{60})$</td>
</tr>
<tr>
<td>64-QAM: Alpha=4</td>
<td>$c = z \times (\sqrt{108})$</td>
</tr>
</tbody>
</table>

3. The soft bits are determined by the decision equations.

$I$ is the real part of product and $Q$ is the imaginary part.

- 64QAM decision equations are:
  
  $b_0 = I$
  
  $b_1 = Q$
  
  $b_2 = 4 - |b_0|$
  
  $b_3 = 4 - |b_1|$
  
  $b_4 = 2 - |b_2|$
  
  $b_5 = 2 - |b_3|$

- 16QAM decision equations are:
  
  $b_0 = I$
  
  $b_1 = Q$
  
  $b_2 = 2 - |b_0|$
  
  $b_3 = 2 - |b_1|$

- QPSK decision equations are:
  
  $b_0 = I$
  
  $b_1 = Q$
The final soft bit information $b$ is the product of soft bits calculated by the above equations and the CSI input.

References


DTV_DQPSKCoder

Description  DQPSK baseband modulator
Library       DTV, Channel Coding
Class         SDFDTV_DQPSKCoder

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Default</th>
<th>Type</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Delay</td>
<td>delay of feedback (as length of register)</td>
<td>384</td>
<td>int</td>
<td>[1, ∞)</td>
</tr>
</tbody>
</table>

Pin Inputs

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>input</td>
<td>input data bits</td>
<td>int</td>
</tr>
</tbody>
</table>

Pin Outputs

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>output</td>
<td>signal after constellation mapping</td>
<td>complex</td>
</tr>
</tbody>
</table>

Notes/Equations

1. This model is used to perform $\pi/4$ shift DQPSK constellation mapping and modulation.
   Each firing, two bits of input data are consumed to produce the complex data output.
2. Phase calculation and mapping is shown in Figure 2-6.
   Complex data is calculated by
(I_j, Q_j) = \begin{bmatrix} \cos \theta_j & -\sin \theta_j \\ \sin \theta_j & \cos \theta_j \end{bmatrix} (I_{j-1}, Q_{j-1})

where (I_j, Q_j) denotes complex data of the j-th symbol and d denotes the number of symbols between (I_j, Q_j) and (I_{j-1}, Q_{j-1}).

<table>
<thead>
<tr>
<th>Input</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>b0*, b1*</td>
<td>\theta_j</td>
</tr>
<tr>
<td>0, 0</td>
<td>\pi/4</td>
</tr>
<tr>
<td>0, 1</td>
<td>-\pi/4</td>
</tr>
<tr>
<td>1, 0</td>
<td>3\pi/4</td>
</tr>
<tr>
<td>1, 1</td>
<td>-3\pi/4</td>
</tr>
</tbody>
</table>

Figure 2-6. Phase Calculation and \pi/4 Shift DQPSK Constellation Mapping

References

DTV_DQPSKDecoder

Description   DQPSK decoder with soft decision
Library       DTV, Channel Coding
Class         SDFDTV_DQPSKDecoder

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Default</th>
<th>Type</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Delay</td>
<td>delay of feedback (as length of register)</td>
<td>384</td>
<td>int</td>
<td>[1, ∞)</td>
</tr>
<tr>
<td>Renorm</td>
<td>option to re-normalize reference phase (set to the nearest symbol point): NO, YES</td>
<td>NO</td>
<td>enum</td>
<td></td>
</tr>
</tbody>
</table>

Pin Inputs

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>input</td>
<td>signal to be demodulated</td>
<td>complex</td>
</tr>
</tbody>
</table>

Pin Outputs

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>output</td>
<td>signal after demodulation</td>
<td>real</td>
</tr>
</tbody>
</table>

Notes/Equations

1. This model is used to perform $\pi/4$ shift DQPSK demodulation; it is the reverse of the process for DTV_DQPSKCoder.

It decodes the complex DQPSK signal to floating-point data to be Viterbi convolutional decoded.

References
Channel Coding Components

**DTV_InnerDecoder**

**Description**  Punctured convolutional decoder

**Library**  DTV, Channel Coding

**Required Licenses**

**Parameters**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Default</th>
<th>Type</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>PuncConvType</td>
<td>punctured convolutional code type: DTV_1_2, DTV_2_3, DTV_3_4, DTV_5_6, DTV_7_8</td>
<td>DTV_1_2</td>
<td>enum</td>
<td></td>
</tr>
<tr>
<td>TrunLen</td>
<td>path memory truncation length (bytes)</td>
<td>10</td>
<td>int</td>
<td>[5, ∞)</td>
</tr>
<tr>
<td>DelayByte</td>
<td>number of bytes to delay for delay adjustment</td>
<td>0</td>
<td>int</td>
<td>[-∞, 204 - TrunLen]</td>
</tr>
</tbody>
</table>

**Pin Inputs**

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>input</td>
<td>signal to be decoded</td>
<td>int</td>
</tr>
</tbody>
</table>

**Pin Outputs**

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>output</td>
<td>decoded signal</td>
<td>int</td>
</tr>
</tbody>
</table>

**Notes/Equations**

1. This subnetwork model performs punctured convolutional decoding over the input signal. The schematic for this subnetwork is shown in Figure 2-7.
Channel Coding Components

![Diagram of DTV_InnerDecoder Schematic]

Figure 2-7. DTV_InnerDecoder Schematic

2. At the receiver side, DTV_PuncDecoder calculates the puncturing pattern and inserts null (zero) at the position where a bit is punctured. A Viterbi decoder is then used to further decode the recovered code bit stream. By replacing the punctured bits with nulls, the punctured bits do not contribute to decoding performance.

The data rate of the mother convolutional code is 1/2.

3. The generator polynomial is $G_1=171_{\text{oct}}$ and $G_2=133_{\text{oct}}$.

References

DTV_InterlvFloat

Description  Interleaver and de-interleaver for float
Library  DTV, Channel Coding
Class  SDFDTV_InterlvFloat

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Default</th>
<th>Type</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Delays</td>
<td>delay of each branch</td>
<td>0 1 2 3</td>
<td>int array</td>
<td>[0, ∞)</td>
</tr>
<tr>
<td>Initial_value</td>
<td>initial value in interleaver</td>
<td>0.0</td>
<td>real</td>
<td>[0, ∞)</td>
</tr>
<tr>
<td>Multiplier</td>
<td>multiple branch number</td>
<td>1</td>
<td>int</td>
<td>[1, ∞)</td>
</tr>
</tbody>
</table>

Pin Inputs

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>input</td>
<td>signal to be interleaved</td>
<td>real</td>
</tr>
</tbody>
</table>

Pin Outputs

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>output</td>
<td>interleaved signal</td>
<td>real</td>
</tr>
</tbody>
</table>

Notes/Equations

1. This model is used to perform floating-point symbol interleaving or de-interleaving over the input signal. It is used to perform de-interleaving of the floating-point symbol after QAM, QPSK or DQPSK decoding.

A general interleaver is used for floating-point data. It is composed of a number of FIFO delay branches as specified by the Delays array in which the delay can...
Channel Coding Components

be specified individually. Multiplier can be used to implement multiple isomorphic interleavers.

The structure of this interleaver is the same as that of DTV_InterlvInt, except this model handles floating-point data.

References


DTV_InterlvInt

Description: Interleaver and de-interleaver for integer
Library: DTV, Channel Coding
Class: SDFDTV_InterlvInt

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Default</th>
<th>Type</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Delays</td>
<td>delay of each branch</td>
<td>0 1 2</td>
<td>int array</td>
<td>[0, ∞)</td>
</tr>
<tr>
<td>Initial_value</td>
<td>initial value in interleaver delay FIFOs</td>
<td>0</td>
<td>int</td>
<td>[0, ∞)</td>
</tr>
<tr>
<td>Multiplier</td>
<td>multiple branch number</td>
<td>1</td>
<td>int</td>
<td>[1, ∞)</td>
</tr>
</tbody>
</table>

Pin Inputs

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>input</td>
<td>signal to be interleaved</td>
<td>int</td>
</tr>
</tbody>
</table>

Pin Outputs

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>output</td>
<td>interleaved signal</td>
<td>int</td>
</tr>
</tbody>
</table>

Notes/Equations

1. This model is used to perform byte-wise interleaving or de-interleaving over the input signal.

A general interleaver is used for integer data. It is composed of a number of FIFO delay branches as specified by the Delays array in which the delay can be specified individually. Multiplier can be used to implement multiple isomorphic interleavers.
2. The conceptual structure of the interleaver used in ISDB-T is described.

Following the conceptual scheme in Figure 2-8, convolutional byte-wise interleaving with length of \( I = 12 \) is applied to the 204-byte packets. For synchronization, the bytes following SYNC bytes will be routed in branch ‘0’ of the interleaver (corresponding to a null delay).

The interleaver can be composed of \( I = 12 \) branches, cyclically connected to the input byte-stream by the input switch. Each branch \( j \) must be a first-in first-out (FIFO) shift register, with length of \( j \times 17 \) bytes. The cells of the FIFO must contain 1 byte, and the input and output switches must be synchronized.

The de-interleaver, in principle, is the same as the interleaver, but the branch indices are reversed.

\( I \) corresponds to the size of Delay array. The size of every branch, \((j \times 17)\) in Figure 2-8, corresponds to the element of Delay array. Multiplier is the number of FIFO register in the interleaver, its value is 1 in Figure 2-8.

The conceptual structure of interleaver with Multiplier = \( N \) and Delay array = \((0, 1 \times K, 2 \times K, \ldots, (I-1) \times K)\) is illustrated in Figure 2-9.

---

![Figure 2-8. Conceptual Diagram of the Interleaver Used in ISDB-T](image-url)
Figure 2-9. Conceptual Diagram of the Interleaver with Multiplier=N and Delay Array=(0, 1×K, 2×K, ..., (I-1)×K)

References


Channel Coding Components

**DTV_Mapper**

**Description**  Uniform and non-uniform mapping for DVB-T and ISDB-T

**Library**  DTV, Channel Coding

**Class**  SDFDTV_Mapper

**Required Licenses**

**Parameters**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Default</th>
<th>Type</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>MappingMode</td>
<td>signal constellations and mapping: QPSK, QAM-16, QAM-64</td>
<td>QAM-16</td>
<td>enum</td>
<td></td>
</tr>
<tr>
<td>Alpha</td>
<td>non-uniform factor for DVB-T.</td>
<td>1</td>
<td>int</td>
<td>[1, ∞)</td>
</tr>
</tbody>
</table>

**Pin Inputs**

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>input</td>
<td>input data bits</td>
<td>int</td>
</tr>
</tbody>
</table>

**Pin Outputs**

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>output</td>
<td>signal after constellation mapping</td>
<td>complex</td>
</tr>
</tbody>
</table>

**Notes/Equations**

1. This model is used to perform QPSK, 16QAM, 64QAM mapping or non-uniform 16QAM and 64QAM Gray mapping.
   
   Each firing, 2 tokens for QPSK, 4 tokens for 16QAM, or 6 tokens for 64QAM are consumed and one constellation symbol is generated.

2. The MappingMode parameter specifies the constellation and mapping mode: QPSK, QAM-16 or QAM-64.
3. The Alpha parameter determines the exact proportions of the constellation. Alpha is the minimum distance separating the two constellation points carrying different high-priority bit values divided by the minimum distance separating any two constellation points. Note that, although the non-hierarchical transmission uses the same uniform constellation as Alpha=1, non-uniform constellation for hierarchical transmission is also supported using Alpha=2 or 4.

4. Bit patterns based on Alpha and MappingMode settings are illustrated in Figure 2-10 through Figure 2-15.

Figure 2-10. QPSK, 16-QAM Mapping Bit Patterns (Non-Hierarchical and Hierarchical, Alpha=1)
Channel Coding Components

Figure 2-11. 64-QAM Mapping Bit Patterns (Non-Hierarchical and Hierarchical, $\alpha=1$)

Figure 2-12. Non-Uniform 16-QAM Mapping Bit Patterns (Alpha=2)
Figure 2-13. Non-Uniform 64-QAM Mapping Bit Patterns (Alpha=2)
Figure 2-14. Non-Uniform 16-QAM Mapping Bit Patterns (Alpha=4)
5. The mapped symbols are normalized; normalization factors are shown in Table 2-5.

**Table 2-5. Normalization Factors**

<table>
<thead>
<tr>
<th>Modulation scheme</th>
<th>Normalization factor</th>
</tr>
</thead>
<tbody>
<tr>
<td>QPSK</td>
<td>$C = \frac{z}{(\sqrt{2})}$</td>
</tr>
<tr>
<td>16-QAM Alpha=1</td>
<td>$C = \frac{z}{(\sqrt{10})}$</td>
</tr>
<tr>
<td>16-QAM Alpha=2</td>
<td>$C = \frac{z}{(\sqrt{20})}$</td>
</tr>
<tr>
<td>16-QAM Alpha=4</td>
<td>$C = \frac{z}{(\sqrt{52})}$</td>
</tr>
<tr>
<td>64-QAM Alpha=1</td>
<td>$C = \frac{z}{(\sqrt{42})}$</td>
</tr>
</tbody>
</table>
Channel Coding Components

### Table 2-5. Normalization Factors

<table>
<thead>
<tr>
<th>Modulation scheme</th>
<th>Normalization factor</th>
</tr>
</thead>
<tbody>
<tr>
<td>64-QAM Alpha=2</td>
<td>$C = z/\sqrt{60}$</td>
</tr>
<tr>
<td>64-QAM Alpha=4</td>
<td>$C = z/\sqrt{108}$</td>
</tr>
</tbody>
</table>

**References**

**DTV_PNreset**

**Description**  PN code source with reset input  
**Library**  DTV, Channel Coding  
**Class**  SDFDTV_PNreset

### Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Default</th>
<th>Type</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Polynomial</td>
<td>generator polynomial (X^0 + X^1 + \ldots + X^M)</td>
<td>1000000000000000</td>
<td>int array</td>
<td>[0, 1]</td>
</tr>
<tr>
<td>Initial</td>
<td>initial and reset value in registers</td>
<td>1001010100000000</td>
<td>int array</td>
<td>[0, 1]</td>
</tr>
<tr>
<td>SignalPoint</td>
<td>output register after each shift</td>
<td>0</td>
<td>int</td>
<td>[0, 31]</td>
</tr>
</tbody>
</table>

### Pin Inputs

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>reset</td>
<td>reset pulse</td>
<td>int</td>
</tr>
</tbody>
</table>

### Pin Outputs

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>output</td>
<td>pseudo random binary sequence</td>
<td>int</td>
</tr>
</tbody>
</table>

### Notes/Equations

1. This model is used to generate a pseudo random binary sequence with less than 32 internal shift registers and an external reset pin.

2. This model generates pseudo random binary sequence for energy dispersal of the TSPs. In the ISDB-T specification, the polynomial for this pseudo random binary sequence generator is:
Channel Coding Components

\[ g(x) = x^{15} + x^{14} + 1 \]

At the start of every OFDM frame, the reset signal becomes valid while the internal shift register is initiated to the sequence of 1001010100000000. PRBS generation is illustrated in Figure 2-16.

![Figure 2-16. PRBS Generation](image)

References


DTV_PuncCoder

Description  Puncture coder
Library  DTV, Channel Coding
Class  SDFDTV_PuncCoder

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Default</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>PuncConvType</td>
<td>punctured convolutional code type: DTV 1/2, DTV 2/3, DTV 3/4, DTV 5/6, DTV 7/8</td>
<td>DTV 1/2</td>
<td>enum</td>
</tr>
</tbody>
</table>

Pin Inputs

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>input</td>
<td>input signal to be perforated</td>
<td>anytype</td>
</tr>
</tbody>
</table>

Pin Outputs

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>output</td>
<td>output signal after perforated</td>
<td>anytype</td>
</tr>
</tbody>
</table>

Notes/Equations

1. This model is used to perforate the input convolutional code to produce a punctured convolutional code. Each firing, K tokens are consumed and N tokens are produced; K and N are determined according to Table 2-6.

2. Punctured convolutional code is usually generated by perforate a mother convolutional code according to a certain pattern to achieve a different data rate. This model determines the perforation pattern according to the code type chosen, then reads the input convolutional coded bits and outputs the input bit or discards it according to the pattern in Table 2-7.
Channel Coding Components

Table 2-6.

<table>
<thead>
<tr>
<th>Code Rate</th>
<th>K (No. of Input Bytes)</th>
<th>N (No. of Output Bits)</th>
</tr>
</thead>
<tbody>
<tr>
<td>DTV 1/2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>DTV 2/3</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>DTV 3/4</td>
<td>6</td>
<td>4</td>
</tr>
<tr>
<td>DTV 5/6</td>
<td>10</td>
<td>6</td>
</tr>
<tr>
<td>DTV 7/8</td>
<td>14</td>
<td>8</td>
</tr>
</tbody>
</table>

Table 2-7. Puncture Pattern and Transmit Sequence

<table>
<thead>
<tr>
<th>Code Rate</th>
<th>Puncture Pattern</th>
<th>Transmit Sequence</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>2/3</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>3/4</td>
<td>6</td>
<td>4</td>
</tr>
<tr>
<td>5/6</td>
<td>10</td>
<td>6</td>
</tr>
<tr>
<td>7/8</td>
<td>14</td>
<td>8</td>
</tr>
</tbody>
</table>

References


**Description**  Punctured convolutional encoder  
**Library**  DTV, Channel Coding  
**Required Licenses**

**Parameters**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Default</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>PuncConvType</td>
<td>punctured convolutional code type: DTV_1_2, DTV_2_3, DTV_3_4, DTV_5_6, DTV_7_8</td>
<td>DTV_1_2</td>
<td>enum</td>
</tr>
</tbody>
</table>

**Pin Inputs**

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>input</td>
<td>signal to be encoded</td>
<td>int</td>
</tr>
</tbody>
</table>

**Pin Outputs**

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>output</td>
<td>encoded signal</td>
<td>int</td>
</tr>
</tbody>
</table>

**Notes/Equations**

1. This subnetwork model performs punctured convolutional encoding. The schematic for this subnetwork is shown in **Figure 2-17**.
2. This subnetwork converts byte input data into bit streams and uses a general convolutional coding model to encode them into mother convolutional code of data rate 1/2.

A puncture encoder generates the punctured convolutional code. PuncConvType specifies the type of DTV punctured convolutional code.

The encoder is illustrated in Figure 2-18.

References

**DTV_PuncConvDecoder**

**Description**  Punctured convolutional decoder  
**Library**  DTV, Channel Coding  

**Parameters**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Default</th>
<th>Type</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>PuncConvType</td>
<td>punctured convolutional code type: DTV_1_2, DTV_2_3, DTV_3_4, DTV_5_6, DTV_7_8</td>
<td>DTV_1_2</td>
<td>enum</td>
<td></td>
</tr>
<tr>
<td>NumBits</td>
<td>number of soft decision bits</td>
<td>4</td>
<td>int</td>
<td>[1, ∞)</td>
</tr>
<tr>
<td>SymbolLen</td>
<td>path memory truncation length</td>
<td>8</td>
<td>int</td>
<td>[1, ∞)</td>
</tr>
<tr>
<td>DelayBit</td>
<td>number of bits to delay for delay adjustment</td>
<td>0</td>
<td>int</td>
<td>[0, ∞]†</td>
</tr>
<tr>
<td>DelayByte</td>
<td>number of bytes to delay for delay adjustment</td>
<td>0</td>
<td>int</td>
<td>[0, 204 - SymbolLen]</td>
</tr>
<tr>
<td>DelayCC</td>
<td>number of bits to delay before Viterbi decoder</td>
<td>0</td>
<td>int</td>
<td>[0, ∞)††</td>
</tr>
</tbody>
</table>

† In ISDB and DVB, DelayBit=0  
†† In ISDB and DVB, DelayCC=0

**Pin Inputs**

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>input</td>
<td>signal to be decoded</td>
<td>int</td>
</tr>
</tbody>
</table>

**Pin Outputs**

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>output</td>
<td>decoded signal</td>
<td>int</td>
</tr>
</tbody>
</table>
Notes/Equations

1. This subnetwork model is used to perform punctured convolutional decoding of data rate 1/2 over the input signal. The schematic for this subnetwork is shown in Figure 2-19.

2. A puncture decoder model is used to decode the punctured convolutional encoded input to normal convolutional coded data. A general Viterbi convolutional decoding model is used to further decode it. After decoding, it performs a bit delay adjustment, packs the bits to bytes and, if necessary, applies a byte delay adjustment before output.

The data rate of the mother convolutional code is 1/2.

The encoder is illustrated in Figure 2-20.

References

Channel Coding Components

**DTV_PuncDecoder**

**Description**  
Puncture decoder

**Library**  
DTV, Channel Coding

**Class**  
SDFDTV_PuncDecoder

### Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Default</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>PuncConvType</td>
<td>Punctured convolution code type: DTV 1/2, DTV 2/3, DTV 3/4, DTV 5/6, DTV 7/8</td>
<td>DTV 1/2</td>
<td>enum</td>
</tr>
</tbody>
</table>

### Pin Inputs

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>input</td>
<td>input signal to be refilled</td>
<td>real</td>
</tr>
</tbody>
</table>

### Pin Outputs

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>output</td>
<td>output signal after refilled</td>
<td>real</td>
</tr>
</tbody>
</table>

### Notes/Equations

1. This model is used to finish inverse procedure of the puncture coder that was perforated during the puncture encoding process. Each firing K tokens are consumed at the input and N tokens are produced according to Table 2-8.

2. This model interpolates a zero value to the punctured data stream to form a full-length data stream. Given the type of DTV punctured convolutional code, it determines the perforation pattern according to PuncConvType, then interpolates zero into the input bits to form the output according to the pattern in Table 2-9.
Table 2-8.

<table>
<thead>
<tr>
<th>PuncConvType</th>
<th>K</th>
<th>N</th>
</tr>
</thead>
<tbody>
<tr>
<td>DTV 1/2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>DTV 2/3</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>DTV 3/4</td>
<td>6</td>
<td>4</td>
</tr>
<tr>
<td>DTV 5/6</td>
<td>10</td>
<td>6</td>
</tr>
<tr>
<td>DTV 7/8</td>
<td>14</td>
<td>8</td>
</tr>
</tbody>
</table>

Table 2-9. Puncture Pattern and Transmit Sequence

<table>
<thead>
<tr>
<th>Code Rate</th>
<th>Puncture Pattern</th>
<th>Transmit Sequence</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>2/3</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>3/4</td>
<td>6</td>
<td>4</td>
</tr>
<tr>
<td>5/6</td>
<td>10</td>
<td>6</td>
</tr>
<tr>
<td>7/8</td>
<td>14</td>
<td>8</td>
</tr>
</tbody>
</table>

References


Channel Coding Components

DTV_QAM16Coder

Description  Uniform and non-uniform 16-QAM encoder for DVB-T and ISDB-T
Library   DTV, Channel Coding
Class   SDFDTV_QAM16Coder

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Default</th>
<th>Type</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alpha</td>
<td>non-uniform factor for DVB-T.</td>
<td>1</td>
<td>int</td>
<td>[1, ∞)</td>
</tr>
</tbody>
</table>

Pin Inputs

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>input</td>
<td>input data bits</td>
<td>int</td>
</tr>
</tbody>
</table>

Pin Outputs

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>output</td>
<td>signal after constellation mapping</td>
<td>complex</td>
</tr>
</tbody>
</table>

Notes/Equations

1. This model is used to perform 16-QAM mapping and non-uniform 16-QAM Gray mapping.

2. This model groups the input data bits to 4-bit groups and maps them into a complex signal from the 16-QAM constellation (illustrated in Figure 2-21), or maps them into a complex signal from the non-uniform 16-QAM constellation, (illustrated in Figure 2-22 and Figure 2-23).
Figure 2-21. 16-QAM Mapping and Corresponding Bit Pattern.
Bit ordering is $y_0, q', y_1, q', y_2, q', y_3, q'$.
This is the constellation of uniform 16-QAM and non-uniform 16-QAM with $\alpha = 1$.

Figure 2-22. Non-uniform 16-QAM with $\alpha = 2$ Mapping and Corresponding Bit Pattern (Bit ordering is the same as Figure 2-21.)
After mapping, the output signal is normalized by a normalization factor $c$, the value of $c$ is:

$$
\begin{align*}
    c &= \frac{z}{\sqrt{10}} & \alpha &= 1 \\
    c &= \frac{z}{\sqrt{20}} & \alpha &= 2 \\
    c &= \frac{z}{\sqrt{52}} & \alpha &= 4
\end{align*}
$$

where $z$ is the complex signal from 16-QAM and non-uniform 16-QAM constellation.

References

Channel Coding Components

DTV_QAM16Decoder

Description: Uniform and non-uniform 16-QAM decoder for DVB-T and ISDB-T
Library: DTV, Channel Coding
Class: SDFDTV_QAM16Decoder

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Default</th>
<th>Type</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alpha</td>
<td>non-uniform factor for DVB-T</td>
<td>1</td>
<td>int</td>
<td>[1, ∞)</td>
</tr>
</tbody>
</table>

Pin Inputs

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>input</td>
<td>signal to be demodulated</td>
<td>complex</td>
</tr>
</tbody>
</table>

Pin Outputs

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>output</td>
<td>signal after demodulation</td>
<td>real</td>
</tr>
</tbody>
</table>

Notes/Equations

1. This model is used to perform 16-QAM de-mapping and non-uniform 16-QAM Gray de-mapping; it is the reverse of the process for DTV_QAM16Coder.

2. This model de-maps the input complex QAM signal data to floating-point data for Viterbi convolutional decoding according to 16-QAM or non-uniform 16-QAM mapping constellation.

References

2-56 DTV_QAM16Decoder

Channel Coding Components

DTV_QAM64Coder

Description Uniform and non-uniform 64-QAM encoder for DVB-T and ISDB-T
Library DTV, Channel Coding
Class SDFDTV_QAM64Coder

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Default</th>
<th>Type</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alpha</td>
<td>non-uniform factor for DVB-T</td>
<td>1</td>
<td>int</td>
<td>[1, ∞)</td>
</tr>
</tbody>
</table>

Pin Inputs

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>input</td>
<td>input data bits</td>
<td>int</td>
</tr>
</tbody>
</table>

Pin Outputs

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>output</td>
<td>signal after constellation mapping</td>
<td>complex</td>
</tr>
</tbody>
</table>

Notes/Equations

1. This model is used to perform 64-QAM mapping and non-uniform 64-QAM Gray mapping.

2. This model groups the input data bits to 6-bits groups and maps them into a complex signal from the 64-QAM constellation (illustrated in Figure 2-24), or maps them into a complex signal from non-uniform 64-QAM constellation (illustrated in Figure 2-25 and Figure 2-26).

After mapping, the output signal is normalized by a normalization factor c, the value of c is:
where $z$ is the complex signal from 64-QAM and non-uniform 64-QAM constellation.

\[
\begin{align*}
C &= \frac{z}{\sqrt{42}}, \alpha = 1 \\
C &= \frac{z}{\sqrt{60}}, \alpha = 2 \\
C &= \frac{z}{\sqrt{108}}, \alpha = 4
\end{align*}
\]

Figure 2-24. 64-QAM mapping and Corresponding Bit Pattern.

Bit order is $Y_0, q', Y_1, q', Y_2, q', Y_3, q', Y_4, q', Y_5, q'$.

This is the constellation of uniform 64-QAM and non-uniform 64-QAM with $\alpha = 1$. 
Channel Coding Components

Figure 2-25. Non-uniform 64-QAM with $\alpha = 2$ mapping and corresponding bit pattern. (Bit order is the same as Figure 2-24.)
Figure 2-26. Non-uniform 64-QAM with $\alpha = 4$ Mapping and Corresponding Bit Pattern (Bit order is the same as Figure 2-24)

References


Channel Coding Components

DTV_QAM64Decoder

Description  Uniform and non-uniform 64-QAM decoder for DVB-T and ISDB-T
Library  DTV, Channel Coding
Class  SDFDTV_QAM64Decoder

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Default</th>
<th>Type</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alpha</td>
<td>non-uniform factor for DVB-T</td>
<td>1</td>
<td>int</td>
<td>[1, ∞)</td>
</tr>
</tbody>
</table>

Pin Inputs

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>input</td>
<td>signal to be demodulated</td>
<td>complex</td>
</tr>
</tbody>
</table>

Pin Outputs

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>output</td>
<td>signal after demodulation</td>
<td>real</td>
</tr>
</tbody>
</table>

Notes/Equations

1. This model is used to perform 64-QAM de-mapping and non-uniform 64-QAM Gray de-mapping; it is the reverse of the process for DTV_QAM64Coder.

2. This model de-maps the input complex QAM signal data to floating-point data for Viterbi convolutional decoding according to 64-QAM or non-uniform 64-QAM mapping constellation.

References

2-62  DTV_QAM64Decoder

Channel Coding Components

**DTV_QPSKCoder**

**Description**  QPSK coder
**Library**  DTV, Channel Coding
**Class**  SDFDTV_QPSKCoder

**Pin Inputs**

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>input</td>
<td>input data bits</td>
<td>int</td>
</tr>
</tbody>
</table>

**Pin Outputs**

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>output</td>
<td>signal after constellation mapping</td>
<td>complex</td>
</tr>
</tbody>
</table>

**Notes/Equations**

1. This model is used to perform QPSK constellation mapping and modulation.
2. This model groups the input data bits to 2-bit groups and maps them into complex data according to the QPSK constellation illustrated in Figure 2-27.

![QPSK Mapping and Corresponding Bit Patterns](image)

Figure 2-27. QPSK Mapping and Corresponding Bit Patterns
References


Channel Coding Components

DTV_QPSKDecoder

Description  QPSK decoder
Library      DTV, Channel Coding
Class        SDFDTV_QPSKDecoder

Pin Inputs

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>input</td>
<td>signal to be demodulated</td>
<td>complex</td>
</tr>
</tbody>
</table>

Pin Outputs

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>output</td>
<td>signal after demodulation</td>
<td>real</td>
</tr>
</tbody>
</table>

Notes/Equations

1. This model is used to perform QPSK demodulation; it is the reverse of the process for DTV_QPSKCoder.

2. This model maps the input complex QPSK signal data to floating-point data for Viterbi convolutional decoding according to the QPSK mapping constellation.

References


DTV_RSCoder

Description   Reed-Solomon encoder
Library       DTV, Channel Coding
Class         SDFDTV_RSCoder

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Default</th>
<th>Sym</th>
<th>Type</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>M</td>
<td>Galois field ((2^M)) size definition</td>
<td>8</td>
<td>M</td>
<td>int</td>
<td>([2, 16])</td>
</tr>
<tr>
<td>K</td>
<td>input code word length</td>
<td>188</td>
<td>K</td>
<td>int</td>
<td>((N-K, N))</td>
</tr>
<tr>
<td>N</td>
<td>output word size (K+parity)</td>
<td>204</td>
<td>N</td>
<td>int</td>
<td>([1, 2^M - 1])</td>
</tr>
<tr>
<td>DataSequence</td>
<td>data array sequence: Forward, Reverse</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SpacePlace</td>
<td>place of zero byte (00h) in data array: Head, Tail</td>
<td>Forward</td>
<td>Head</td>
<td>enum</td>
<td></td>
</tr>
<tr>
<td>ParitySequence</td>
<td>parity array sequence: forward, reverse</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Encode47H</td>
<td>place 47H in data array: yes, no</td>
<td>yes</td>
<td></td>
<td>enum</td>
<td></td>
</tr>
</tbody>
</table>

Pin Inputs

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>input</td>
<td>signal to be encoded</td>
<td>int</td>
</tr>
</tbody>
</table>

Pin Outputs

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>output</td>
<td>error protected signal</td>
<td>int</td>
</tr>
</tbody>
</table>

Notes/Equations
Channel Coding Components

1. This model is used to perform shortened Reed-Solomon error correcting encoding over the input signal for DVB-T and ISDB-T systems. Each firing, K tokens are consumed at the input pin and N tokens are produced. For the shortened code, \((2^M - 1 - N)\) zero symbols are added in the information symbols and N-K parity symbols are generated. N-K parity symbols are appended to the input signal to form the output.

If DataSequence = Forward, the order of symbols to be encoded is the same as their input order, otherwise the order is the reverse of their input order.

If SpacePlace = Head, zero symbols are added before the information symbols, otherwise they are added behind the information symbols.

If ParitySequence = forward, the output order of parity symbols is the same as the original order when they are generated, otherwise, the order is reverse of the original order.

If Encode47H = yes, the first byte 47H, the sync byte of every Transport Stream Packet (TSP) is treated as the information symbol and encoded, otherwise, it is not encoded and the number of zero symbols added is changed to \((2^M - N)\).

Values in DVB-T and ISDB-T are: DataSequence=Forward, SpacePlace=Head, ParitySequence=forward, Encode47H =yes.

2. The value range of the input data should be \([0, 2^M - 1]\).

3. Galois field generator polynomial is automatically selected according to the value of M.

4. Implementation

The code format is: RS code \((n, k)\), defined on Galois Field \((2^m)\).

**Galois Field Generator**

Galois fields are set up depending on the number of bits per symbol and the number of symbols per block.

**Algorithm**

Generate GF \((2^m)\) from the irreducible primitive polynomial. It is defined as the polynomial of least degree, with coefficients in GF(2) and a highest degree coefficient equal to 1. The polynomial is always degree m.
The elements of Galois field can have two representations: exponent or polynomial. Let $\alpha$ represent the root of the primitive polynomial $p(x)$. Then in $\text{GF}(2^m)$, for any $0 \leq i \leq 2^m - 2$

$$\alpha^i = b_i(0) + b_i(1)\alpha + b_i(2)\alpha^2 + \cdots + b_i(m-1)\alpha^{m-1}$$

where the binary vector $(b_i(0), b_i(1), \ldots, b_i(m-1))$ is the representation of the integer polynomial $[i]$. Now $\text{exponent}[i]$ is the element whose polynomial representation is $(b_i(0), b_i(1), \ldots, b_i(m-1))$, and $\text{exponent}[\text{polynomial}[i]] = i$.

The polynomial representation is convenient for addition while the exponent representation for multiplication.

**RS Encoder**

The RS generator polynomial is more generally defined as

$$g(x) = (x - a^{m_0})(x - a^{m_1})\cdots(x - a^{m_{2t-1}})$$

where $t$ is the correctable error number. It can be reduced to a $2t$ order of polynomial

$$g(x) = x^{2t} + g_{2t-1}x^{2t-1} + \ldots + g_0$$

The encoding is done by using a feedback shift register with appropriate connections specified by the element $g_i$. The encoded symbol is then

$$\text{in}(x) \times x^{(n-k)} + \text{parity}(x)$$

where $\text{in}(x)$ is the polynomial representation of the input data, $\text{parity}(x)$ is the polynomial of the parity symbol.

The RS encoder diagram is shown in Figure 2-28.
5. Field generator polynomial of DVB-T is
\[ p(x) = x^8 + x^4 + x^3 + x^2 + 1 \]

The shortened Reed-Solomon code is implemented by adding 51 bytes, all set to zero, before the information bytes at the input of an RS (255,239, t = 8) encoder. After RS coding, these null bytes are discarded, leading to an RS code word of N=204 bytes.

References


### Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Default</th>
<th>Sym</th>
<th>Type</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>M</td>
<td>Galois field (2^M) size definition</td>
<td>8</td>
<td>m</td>
<td>int</td>
<td>([2, 16])</td>
</tr>
<tr>
<td>K</td>
<td>size of output block (data)</td>
<td>188</td>
<td>k</td>
<td>int</td>
<td>((N-K, N))</td>
</tr>
<tr>
<td>N</td>
<td>size of input block (data + parity)</td>
<td>204</td>
<td>n</td>
<td>int</td>
<td>([1, 2^M - 1])</td>
</tr>
<tr>
<td>Error</td>
<td>stop simulation option</td>
<td>0</td>
<td>int</td>
<td></td>
<td>([0, \infty])†</td>
</tr>
<tr>
<td>DataSequence</td>
<td>data array sequence: Forward, Reverse</td>
<td></td>
<td></td>
<td>enum</td>
<td></td>
</tr>
<tr>
<td>SpacePlace</td>
<td>place of zero byte (00h) in data array: Head, Tail</td>
<td></td>
<td></td>
<td>enum</td>
<td></td>
</tr>
<tr>
<td>ParitySequence</td>
<td>parity array sequence: forward, reverse</td>
<td></td>
<td></td>
<td>enum</td>
<td></td>
</tr>
<tr>
<td>Encode47H</td>
<td>place 47H in data array: yes, no</td>
<td></td>
<td></td>
<td>enum</td>
<td></td>
</tr>
</tbody>
</table>

† when Error> 0, simulation will stop if the number of received uncorrectable RS packets is larger than the number specified.

### Pin Inputs

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>input</td>
<td>signal to be decoded</td>
<td>int</td>
</tr>
</tbody>
</table>

### Pin Outputs

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>output</td>
<td>decoded signal</td>
<td>int</td>
</tr>
</tbody>
</table>
Notes/Equations

1. This model is used to perform Reed-Solomon error correcting decoding over the input signal. Each firing, N tokens are consumed at the input port and K tokens are produced.

If DataSequence = Forward, the order of symbols to be decoded is the same as their input order, otherwise the order is the reverse of their input order.

If SpacePlace = Head, \((2^M-1-N)\) zero symbols are added before the information symbols before decoding, otherwise they are added behind the information symbols.

If ParitySequence = forward, the input order of parity symbols is the same as the order needed in decoding, otherwise, the order is reverse of the decoding order.

If Encode47H = yes, the first byte 47H, the sync byte of every Transport Stream Packet (TSP) is treated as the information symbol and decoded, otherwise, it is not decoded and the number of zero symbols added is changed to \((2^M-N)\).

Values in DVB-T and ISDB-T are: DataSequence=Forward, SpacePlace=Head, ParitySequence=forward, Encode47H =yes.

2. Galois field generator polynomial is automatically selected according to the value of M.

3. Implementation

Decoding Routines
For the shortened code, the same number of zero symbols is inserted into the same position as in the RS encoder, and a Reed-Solomon decoder is used to decode the block. After decoding, the padded symbols are discarded leaving the desired information symbols.

Getting Syndromes
Syndromes indicate an erroneous situation. When the generator polynomial \(g(x)\) and the received code word \(r(x)\) are given, the occurrence of one or more errors during transmission of an encoded block is known.

Let

\[
v(x) = v_0 + v_1 x + v_2 x^2 + \ldots + v_{n-1} x^{n-1}
\]
where \( v(x) \) is the polynomial representation of the transmitted symbol.

\[
\begin{align*}
  r(x) &= r_0 + r_1 x + r_2 x^2 + \ldots + r_{n-1} x^{n-1} \\
\end{align*}
\]

where \( r(x) \) is the polynomial representation of the received symbol.

Then

\[
  r(x) = v(x) + e(x)
\]

where \( e(x) \) denotes the error patterns. If \( r_i = v_i \), then \( e_i = 0 \); else \( e_i = 1 \).

Remember,

\[
  v(x) = g(x)Q(x)
\]

where \( Q(x) \) is the quotient.

So, if \( \alpha^i \) is the root of \( g(x) \), then

\[
  v(\alpha^i) = 0 \quad \text{and} \quad r(\alpha^i) = e(\alpha^i)
\]

To check the occurrence of errors at the receiver, calculate syndromes \( s(i) \); the syndromes are determined by the error patterns

\[
  s(i) = e(\alpha^{m_b + i})
\]

If one or more of the syndromes are not equal to 0, one or more symbol errors occur in the received data. For example, if

\[
\alpha^{m_b}, \alpha^{m_b+1}, \ldots, \alpha^{m_b+2t-1}
\]

are roots of \( g(x) \), then

\[
\begin{align*}
  s(1) &= r(\alpha^{m_b}) \\
  s(2) &= r(\alpha^{m_b+1}) \\
  \vdots \vdotwith{20pt} \\
  s(2t) &= r(\alpha^{m_b+2t-1})
\end{align*}
\]
Use the syndromes to find the error location polynomial.

Given the syndromes \( s(i) \), the decoding algorithm will synthesize an error location polynomial. The roots of the polynomial indicate the error positions.

Assuming the received symbols have \( v \) symbol errors, the syndromes are represented as follows.

\[
\begin{align*}
    s(1) &= \beta_1^{m_0} + \beta_2^{m_0} + \ldots + \beta_v^{m_0} \\
    s(2) &= \beta_1^{m_0 + 1} + \beta_2^{m_0 + 1} + \ldots + \beta_v^{m_0 + 1} \\
    &
\end{align*}
\]

where

\[
    \beta_l = a_l
\]

and

\[
    a_l \quad (1 \leq l \leq v)
\]

is the error location and \( m_0 \) can be any integer. (Generally, \( m_0 \) is 0 or 1.)

Now the error location polynomial is defined as

\[
\begin{align*}
    \Omega(x) &= (1 + \beta_1^{m_0} x)(1 + \beta_2^{m_0} x)\ldots(1 + \beta_v^{m_0} x) \\
    &= \Omega_0 + \Omega_1 x + \Omega_2 x^2 + \ldots + \Omega_v x^v
\end{align*}
\]

The Berlekamp iterative algorithm is used to construct this polynomial, which is the key to RS decoding. (The algorithm is described here without proof. For more information, refer to [3].) An iterative table will be filled as shown in Table 2-10.
where $\mu$ is the iterative step number, $d_\mu$ is the $\mu$-th step iterative difference, $l_\mu$ is the order of $\Omega^{(\mu)}(x)$.

- If $d_\mu = 0$
  
  then $\Omega^{(\mu+1)}(x) = \Omega^{(\mu)}(x)$ and $l_{\mu+1} = l_\mu$.

- If $d_\mu \neq 0$, search for lines in the table to find step $\rho$ in which $d_\rho \neq 0$ and the value of $\rho - l_\rho$ is the maximum,

  then

  $\Omega^{(\mu+1)}(x) = \Omega^{(\mu)}(x) - d_\mu d_\rho^{-1} x^{(\mu-\rho)} \Omega^{(\rho)}(x)$

  and

  $l_{\mu+1} = \max(l_\mu, l_\rho + \mu - \rho)$.

For the two conditions

$$d_{\mu+1} = s_{\mu+2} + \Omega^{(\mu+1)}_{\mu+1} s_{\mu+1} + \ldots + \Omega^{(\mu+1)}_{l_{\mu+1}} s_{\mu+2 - l_{\mu+1}}$$

Iterate until the last line of the table $\Omega^{(2t)}(x)$ is determined. If the order of the polynomial is greater than $t$, which means the received codeword block has more than $t$ errors, the error cannot be corrected.

Determining Error Values
Channel Coding Components

In the case of non-binary codes, error values must be known. Error values will be solved and corrected, unless the order of the error location polynomial is greater than \( t \), in which case uncorrected information symbols will not be used.

The minimum order polynomial is found by iterating and solved to obtain the least number of roots (error location number). The inverse element of the root indicates the error location.

The error value is determined by the equation from [3]:

\[
e_j = \beta_j^{(1-m_v)} \frac{z(\beta_1^{-1})}{\prod_{i=1}^{v} (1 + \beta_i \beta_1^{-1})}
\]

where

\[
z(x) = 1 + (s_1 + \Omega_1)x + (s_2 + \Omega_1 s_1 + \Omega_2)x^2 + \ldots + (s_v + \Omega_1 s_{v-1} + \Omega_2 s_{v-2} + \ldots + \Omega_v)x^v
\]

then

\[
\text{out}(x) = r(x) - e(x).
\]

4. The field generator polynomial of DVB-T and ISDB-T is

\[
p(x) = x^8 + x^4 + x^3 + x^2 + 1
\]

The shortened RS code is implemented by adding 51 bytes, all set to zero, before the information bytes at the input of an RS \((255,239, t = 8)\) encode. After RS decoding, these null bytes are discarded, leading to an RS decode word of \( K=188\) bytes.

References


Channel Coding Components

**DTV_ScrambleByte**

**Description**  Byte scrambler

**Library**    DTV, Channel Coding

**Required Licenses**

**Parameters**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Default</th>
<th>Type</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Delay</td>
<td>number of bytes to delay for delay adjustment</td>
<td>8</td>
<td>int</td>
<td>[0, ∞)</td>
</tr>
</tbody>
</table>

**Pin Inputs**

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>input</td>
<td>signal to be encoded</td>
<td>int</td>
</tr>
</tbody>
</table>

**Pin Outputs**

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>output</td>
<td>encoded signal</td>
<td>int</td>
</tr>
</tbody>
</table>

**Notes/Equations**

1. This subnetwork model is used to randomize the MPEG-2 transport MUX packet, which is 188 bytes.

   The schematic for this subnetwork is shown in Figure 2-29.
2. The polynomial for the pseudo random binary sequence generator is:

\[ g(x) = x^{15} + x^{14} + 1 \]

The first one sync-word byte is unrandomized. Loading of the sequence 1001010100000000 into the PRBS registers is initiated at the start of each of the 8 transport packets.

References

Channel Coding Components
Chapter 3: DVB-T Components
DVB-T Components

**DTV_CIRNorm**

**Description** Soft output for soft decision decoder

**Library** DTV, DVB-T

**Class** SDFDTV_CIRNorm

**Required Licenses**

<table>
<thead>
<tr>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Name</strong></td>
</tr>
<tr>
<td>Mode</td>
</tr>
<tr>
<td>SoftDecision</td>
</tr>
</tbody>
</table>

**Pin Inputs**

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>input</td>
<td>channel coefficient in active subcarriers</td>
<td>complex</td>
</tr>
</tbody>
</table>

**Pin Outputs**

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>output</td>
<td>channel status information</td>
<td>real</td>
</tr>
</tbody>
</table>

**Notes/Equations**

1. This model is used to output soft information for soft decision decoder. The input signal is channel coefficient in active subcarriers \(= H(i)\).
   - When SoftDecision=Hard, the output signal is 1.0 in all active subcarriers.
   - When SoftDecision=Soft, the output signal is CSI \(= |H(i)|^2 / \text{Max}(|H(i)|^2)\) which is limited to \([0, 1]\).
• When SoftDecision=Cochannel, the 1.0 output in the subcarrier signal is slightly affected by the PAL co-channel analog signal; 0.0 in other subcarriers is severely affected by the PAL co-channel analog signal.

The PAL analog signal bandwidth is 8 MHz.

The number of tokens each firing is 1512 for 2k mode and 6048 for 8k mode.

References

DVB-T Components

**DTV_DVBBitBlockInterlv**

**Description**  DVB bit interleaver  
**Library**  DTV, DVB-T  
**Class**  SDFDTV_DVBBitBlockInterlv

### Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Default</th>
<th>Type</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>BlockSize</td>
<td>size of bit block to be interleaved</td>
<td>126</td>
<td>int</td>
<td>(0, ∞)</td>
</tr>
<tr>
<td>Offset</td>
<td>offset of permutation in interleaver</td>
<td>0</td>
<td>int</td>
<td>(0, BlockSize)</td>
</tr>
<tr>
<td>Option</td>
<td>operation option: Interleaving, De-interleaving</td>
<td>Interleaving</td>
<td>enum</td>
<td></td>
</tr>
</tbody>
</table>

### Pin Inputs

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>input</td>
<td>input data bits</td>
<td>anytype</td>
</tr>
</tbody>
</table>

### Pin Outputs

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>output</td>
<td>interleaved OFDM symbols</td>
<td>anytype</td>
</tr>
</tbody>
</table>

### Notes/Equations

1. This model is used to perform bit block interleaving in DVB-T systems. The bit interleaving block size is 126 bits in DVB-T systems. The block interleaving process is repeated 12 times per OFDM symbol of useful data in the 2k mode and 48 times per OFDM symbol in the 8k mode.
Each firing, this model consumes BlockSize points of input data and produces the same amount of output data.

References

DVB-T Components

**DTV_DVBChannel**

**Description**  
DVB channel model for transmission

**Library**  
DTV, DVB-T

**Class**  
SDFDTV_DVBChannel

**Parameters**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Default</th>
<th>Unit</th>
<th>Type</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>ChannelModel</td>
<td>test path environment: Fixed Reception F1, Portable Reception P1</td>
<td>Fixed Reception F1</td>
<td>enum</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SampleTime</td>
<td>time interval per symbol</td>
<td>0.123046875e-6 sec</td>
<td>sec</td>
<td>real</td>
<td>(0, ∞)</td>
</tr>
</tbody>
</table>

**Pin Inputs**

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>in</td>
<td>input signal to channel</td>
<td>complex</td>
</tr>
</tbody>
</table>

**Pin Outputs**

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>out</td>
<td>output signal from channel</td>
<td>complex</td>
</tr>
</tbody>
</table>

**Notes/Equations**

1. This model is a multipath fading channel with 20 reflected paths to simulate the baseband. It is described in DVB-T specification for performance evaluation.

2. The channel models were generated based on the following equations where \(x(t)\) and \(y(t)\) are input and output signals, respectively.

**Fixed Reception F1**

The relation between the output signal \(y(t)\) and input signal \(x(t)\) is:
where

• the first term before the sum represents the line of sight ray

• $N$ is the number of echoes equal to 20

• $\theta_i$ is the phase shift from scattering of the $i$-th path (listed in Table 3-1)

• $\rho_i$ is the attenuation of the $i$-th path (listed in Table 3-1)

• $\tau_i$ is the relative delay of the $i$-th path (listed in Table 3-1)

The Rician factor $K$ (ratio of line of sight ray to reflected paths) is:

$$K = \frac{\rho_0^2}{\sum_{i=1}^{N} \rho_i^2}$$

In simulations, a Rician factor $K=10$ dB has been used:

$$\rho_0 = \sqrt{10 \sum_{i=1}^{N} \rho_i^2}$$

**Portable Reception P1, Rayleigh fading**

$$y(t) = k \sum_{i=1}^{N} \rho_i e^{-j2\pi \theta_i} \times (t - \tau_i)$$

where
DVB-T Components

\[ k = \frac{1}{\sqrt{\sum_{i=1}^{N} \rho_i^2}} \]

Table 3-1. Attenuation, Phase and Delay Values for F1 and P1

<table>
<thead>
<tr>
<th>(i)</th>
<th>(\rho_i)</th>
<th>(\tau_i[\text{us}])</th>
<th>(\theta_i[\text{rad}])</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.057662</td>
<td>1.003019</td>
<td>4.855121</td>
</tr>
<tr>
<td>2</td>
<td>0.176809</td>
<td>5.422091</td>
<td>3.419109</td>
</tr>
<tr>
<td>3</td>
<td>0.407163</td>
<td>0.518650</td>
<td>5.864470</td>
</tr>
<tr>
<td>4</td>
<td>0.303585</td>
<td>2.751772</td>
<td>2.215894</td>
</tr>
<tr>
<td>5</td>
<td>0.258782</td>
<td>0.602895</td>
<td>3.758058</td>
</tr>
<tr>
<td>6</td>
<td>0.061831</td>
<td>1.016585</td>
<td>5.430202</td>
</tr>
<tr>
<td>7</td>
<td>0.150340</td>
<td>0.143556</td>
<td>3.952093</td>
</tr>
<tr>
<td>8</td>
<td>0.051534</td>
<td>0.513832</td>
<td>1.093586</td>
</tr>
<tr>
<td>9</td>
<td>0.185074</td>
<td>3.324866</td>
<td>5.775198</td>
</tr>
<tr>
<td>10</td>
<td>0.400967</td>
<td>1.935570</td>
<td>0.154459</td>
</tr>
<tr>
<td>11</td>
<td>0.295723</td>
<td>0.429948</td>
<td>5.928333</td>
</tr>
<tr>
<td>12</td>
<td>0.350825</td>
<td>3.228872</td>
<td>3.053023</td>
</tr>
<tr>
<td>13</td>
<td>0.262909</td>
<td>0.848831</td>
<td>0.628578</td>
</tr>
<tr>
<td>14</td>
<td>0.225894</td>
<td>0.073883</td>
<td>2.128544</td>
</tr>
<tr>
<td>15</td>
<td>0.170996</td>
<td>0.203952</td>
<td>1.099463</td>
</tr>
<tr>
<td>16</td>
<td>0.149723</td>
<td>0.194207</td>
<td>3.462951</td>
</tr>
<tr>
<td>17</td>
<td>0.240140</td>
<td>0.924450</td>
<td>3.664773</td>
</tr>
<tr>
<td>18</td>
<td>0.116587</td>
<td>1.381320</td>
<td>2.833799</td>
</tr>
<tr>
<td>19</td>
<td>0.221155</td>
<td>0.640512</td>
<td>3.334290</td>
</tr>
<tr>
<td>20</td>
<td>0.259730</td>
<td>1.368671</td>
<td>0.393889</td>
</tr>
</tbody>
</table>

References

DTV_DVBChEstimator

Description  Linear channel estimator and channel interpolator for DVB-T
Library  DTV, DVB-T
Class  SDFDTV_DVBChEstimator

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Default</th>
<th>Type</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Carriers</td>
<td>number of carriers in one OFDM symbol</td>
<td>1705</td>
<td>int</td>
<td>1705 for 2k mode, 6817 for 8k mode</td>
</tr>
<tr>
<td>Order</td>
<td>FFT points=2^Order</td>
<td>11</td>
<td>int</td>
<td>(0, ∞)†</td>
</tr>
<tr>
<td>SPperiod</td>
<td>distance in carriers between nearby scattered pilots</td>
<td>12</td>
<td>int</td>
<td>[0, ∞)††</td>
</tr>
<tr>
<td>SPstart</td>
<td>start position of scattered pilots in carriers</td>
<td>0</td>
<td>int</td>
<td>[0, ∞)††</td>
</tr>
<tr>
<td>SPoffset</td>
<td>offset value of scattered pilots in each symbol</td>
<td>3</td>
<td>int</td>
<td>[0, ∞)††</td>
</tr>
<tr>
<td>SPphase</td>
<td>initial phase of scattered pilots</td>
<td>0</td>
<td>int</td>
<td>[0, SPperiod/SPoffset-1)†††</td>
</tr>
</tbody>
</table>

† Order is the order of FFT; it must satisfy 2^Order ≥ Carriers
†† In DVB-T systems: SPperiod=12, SPstart=0, SPoffset=3
††† SPphase=3 in OFDM receiver (because DTV_MLEstimator makes one OFDM symbol delay).

Pin Inputs

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>input</td>
<td>output signals from FFT</td>
<td>complex</td>
</tr>
</tbody>
</table>
DVB-T Components

Pin Outputs

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>output</td>
<td>signals in active subcarriers</td>
<td>complex</td>
</tr>
<tr>
<td>3</td>
<td>Coef</td>
<td>channel coefficient in active subcarriers</td>
<td>complex</td>
</tr>
</tbody>
</table>

Notes/Equations

1. The model provides linear channel estimation and channel interpolation based on the pilot channel; data from the active subcarriers is output.

2. The number of IntState of Length determines the number of CP (continual pilot) and TPS in each symbol; the position of CP and TPS are determined according to Table 3-2 and Table 3-3, respectively.

Table 3-2. Carrier Indices for Continual Pilot Carriers

<table>
<thead>
<tr>
<th>Continual pilot carrier positions (index number k)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2k mode</td>
</tr>
<tr>
<td>0 48 54 87 141 156 192 201 255 279 282 333 432 450 483 525</td>
</tr>
<tr>
<td>531 618 636 714 759 765 780 804 873 888 918 939 942 969 984</td>
</tr>
<tr>
<td>1050 1101 1107 1110 1137 1140 1146 1206 1269 1323 1377 1491</td>
</tr>
<tr>
<td>1683 1704</td>
</tr>
<tr>
<td>8k mode</td>
</tr>
<tr>
<td>0 48 54 87 141 156 192 201 255 279 282 333 432 450 483 525</td>
</tr>
<tr>
<td>531 618 636 714 759 765 780 804 873 888 918 939 942 969 984</td>
</tr>
<tr>
<td>1050 1101 1107 1110 1137 1140 1146 1206 1269 1323 1377 1491</td>
</tr>
<tr>
<td>1683 1704 1752 1758 1791 1845 1860 1896 1905 1959 1983 1986</td>
</tr>
<tr>
<td>2037 2136 2154 2187 2229 2235 2322 2340 2418 2463 2469 2484</td>
</tr>
<tr>
<td>2508 2577 2592 2622 2643 2646 2673 2688 2754 2805 2811 2814</td>
</tr>
<tr>
<td>2841 2844 2850 2910 2953 3027 3081 3195 3387 3408 3456 3462</td>
</tr>
<tr>
<td>3493 3539 3543 3600 3609 3663 3687 3690 3741 3840 3858 3891</td>
</tr>
<tr>
<td>3933 3939 4026 4044 4122 4167 4173 4186 4212 4281 4296 4326</td>
</tr>
<tr>
<td>4347 4350 4377 4392 4458 4509 4515 4518 4545 4548 4554 4614</td>
</tr>
<tr>
<td>4677 4731 4785 4899 5091 5112 5160 5166 5199 5253 5268 5304</td>
</tr>
<tr>
<td>5313 5367 5391 5394 5445 5544 5562 5595 5637 5643 5730 5748</td>
</tr>
<tr>
<td>5826 5871 5877 5892 5916 5985 6000 6051 6054 6081 6096 6162</td>
</tr>
<tr>
<td>6213 6219 6222 6249 6252 6258 6318 6381 6435 6489 6603 6795</td>
</tr>
<tr>
<td>6816</td>
</tr>
</tbody>
</table>

Table 3-3. Carrier Indices for TPS Carriers

<table>
<thead>
<tr>
<th>TPS carriers (index number k)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2k mode</td>
</tr>
<tr>
<td>34 50 209 346 413 569 595 688 790 901 1073 1219 1262 1286 1469 1594 1687</td>
</tr>
<tr>
<td>8k mode</td>
</tr>
<tr>
<td>34 50 209 346 413 569 595 688 790 901 1073 1219 1262 1286 1469 1594 1687 1738 1754 1913 2050 2117 2273 2299 2392 2494 2605 2777 2923 2966 2990 3173 3298 3391 3442 3458 3617 3754 3821 3977 4003 4096 4198 4309 4481 4627 4670 4694 4877 5002 5095 5146 5162 5321 5458 5525 5681 5707 5800 5902 6013 6185 6331 6374 6398 6581 6706 6799</td>
</tr>
</tbody>
</table>

PRBS sequences in each segment are generated according to Figure 3-1; the initial sets of PRBS register is 111111111111. The PRBS is initialized so that the
first output bit from the PRBS coincides with the first active carrier. A new
value is generated by the PRBS on each carrier used in each segment (whether
or not it is a pilot).

Positions of the corresponding scattered pilots are generated as follows. For the
symbol of index \( l \) (ranging from 0 to 67), carriers for which index \( k \) belongs to
subset

\[
\{ k = K_{\min} + 3 \times l \mod 4 + 12 \ p \ \text{integer}, \ p \geq 0, \ k \in [K_{\min}, K_{\max}] \}
\]

are scattered pilot positions.

where \( p \) is an integer that takes all possible values \( \geq 0 \), provided the resulting
value for \( k \) does not exceed the valid range \([K_{\min}, K_{\max}]\); \( K_{\min}=0 \), and
\( K_{\max}=1704 \) for 2k mode or 6816 for 8k mode. SP_\text{period}, SP_\text{start}, SP_\text{offset}
and SP_\text{phase} parameters control the scattered pilots positions.

After determining all CP, TPS, and SP positions in each symbol and the value of
the PRBS sequence in all active carriers in symbol, this model demultiplexes
input data into TPS and TSP data. According to the TPS position and the PRBS
sequence, one TPS bit in each TPS position is output.

After determining all CP and SP positions in OFDM symbol, we get the pilot
value from the PRBS sequence; channel estimation in CP and SP pilots can be
determined:

\[
h(i) = \frac{x(i)}{\text{PilotValue}(i)}
\]

where \( h(i) \) is the channel estimation, \( x(i) \) is the received signal from channel
after FFT, and \( \text{PilotValue}(i) \) is the PRBS value corresponding to CP and SP
positions in OFDM symbol.
DVB-T Components

From the subchannel estimation in the CP and SP pilots position, we use the linear interpolation algorithm for active subchannel estimations as follows:

\[ h(k) = \alpha h(i) + (1 - \alpha) h(j) \]

\[ \alpha = \frac{j - k}{j - i} \]

where \( i \leq h \leq j \), \( h(i) \), and \( h(j) \) are the subchannel estimations of CP or SP pilots. The active carriers and corresponding subchannel carriers are output for use in the equalizer model. The inserted zeros in DTV_DVBLoadIFFTBuff are discarded.

References

DTV_DVB2DChEstimator

Description  Channel estimator and two-dimension channel interpolator for DVB-T
Library  DTV, DVB-T
Class  SDFDTV_DVB2DChEstimator
Required Licenses

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Default</th>
<th>Type</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Carriers</td>
<td>number of carriers in one OFDM symbol</td>
<td>1705</td>
<td>int</td>
<td>1705 for 2k mode, 6817 for 8k mode</td>
</tr>
<tr>
<td>Order</td>
<td>FFT points=2^Order</td>
<td>11</td>
<td>int</td>
<td>(0, ∞)†</td>
</tr>
<tr>
<td>SPeriod</td>
<td>distance in carriers between nearby scattered pilots</td>
<td>12</td>
<td>int</td>
<td>(0, ∞)††</td>
</tr>
<tr>
<td>SStart</td>
<td>start position of scattered pilots in carriers</td>
<td>0</td>
<td>int</td>
<td>(0, ∞)††</td>
</tr>
<tr>
<td>SOffset</td>
<td>offset value of scattered pilots in each symbol</td>
<td>3</td>
<td>int</td>
<td>(0, ∞)††</td>
</tr>
<tr>
<td>SPhase</td>
<td>initial phase of scattered pilots</td>
<td>0</td>
<td>int</td>
<td>[0, SPeriod/ SOffset-1]†††</td>
</tr>
</tbody>
</table>

† Order is the order of FFT; it must satisfy 2^Order ≥ Carriers
†† In DVB-T systems: SPeriod=12, SStart=0, SOffset=3
††† SPhase=0 in OFDM receiver.

Pin Inputs

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>input</td>
<td>output signals from FFT</td>
<td>complex</td>
</tr>
</tbody>
</table>
DVB-T Components

Pin Outputs

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>output</td>
<td>signals in active subcarriers</td>
<td>complex</td>
</tr>
<tr>
<td>3</td>
<td>Coef</td>
<td>channel coefficient in active subcarriers</td>
<td>complex</td>
</tr>
</tbody>
</table>

Notes/Equations

1. The model provides linear channel estimation and 2D channel interpolation using collected scattered pilots; data from the active subcarriers is output.

2. The number of IntState of Length determines the number of CP (continual pilot) and TPS in each symbol; the position of CP and TPS are determined according to Table 3-4 and Table 3-5, respectively.

PRBS sequences in each segment are generated according to Figure 3-2; the initial sets of PRBS register is 11111111111. The PRBS is initialized so that the first output bit from the PRBS coincides with the first active carrier. A new value is generated by the PRBS on every used carrier in each segment (whether or not it is a pilot).

Positions of the corresponding scattered pilots are generated as follows. For the symbol of index \( l \) (ranging from 0 to 67), carriers for which index \( k \) belongs to subset

\[
\{ k = K_{\min} + 3 \times l \mod 4 + 12 p \mid p \text{ integer}, p \geq 0, k \in [K_{\min}, K_{\max}] \}
\]

are scattered pilot positions.

where \( p \) is an integer that takes all possible values \( \geq 0 \), provided the resulting value for \( k \) does not exceed the valid range \([K_{\min}, K_{\max}]\); \( K_{\min}=0 \), and \( K_{\max}=1704 \) for 2k mode or 6816 for 8k mode. SPperiod, SPstart, SPOffset and SPPhase parameters control the scattered pilots positions.
Table 3-4. Carrier Indices for Continual Pilot Carriers

<table>
<thead>
<tr>
<th>Continual pilot carrier positions (index number k)</th>
<th>2k mode</th>
<th>8k mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 48 54 87 141 156 192 201 255 279 282 333 432</td>
<td>450 483</td>
<td>525 531</td>
</tr>
<tr>
<td>636 714 759 765 780 804 873 888 918 939 942</td>
<td>946 969</td>
<td>984 1008</td>
</tr>
<tr>
<td>946 1008</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 3-5. Carrier Indices for TPS Carriers

<table>
<thead>
<tr>
<th>2k mode</th>
<th>8k mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>34 50 209 346 413 569 595 688 790 901 1073</td>
<td>1219 1262 1286 1469 1594 1687</td>
</tr>
<tr>
<td>1219 1262 1286 1469 1594 1687</td>
<td></td>
</tr>
</tbody>
</table>

Initialization sequence

1-bit delay

1-bit delay

1-bit delay

1-bit delay

1-bit delay

1-bit delay

1-bit delay

1-bit delay

1-bit delay

PRBS sequence starts: 1111...
pilots positions in four OFDM symbols, we get the pilot values from the PRBS sequence; channel estimation in scattered pilots can be determined by:

\[ h(i) = \frac{x(i)}{\text{PilotValue}(i)} \]

where \( h(i) \) is the channel estimation, \( x(i) \) is the received signal from channel after FFT, and \( \text{PilotValue}(i) \) is the PRBS value corresponding to CP and SP positions in OFDM symbol.

From the subchannel estimation in the scattered pilots positions, we can use the linear interpolation algorithm for active subchannel estimations as follows:

\[ h(k) = \alpha h(i) + (1 - \alpha) h(j) \]

\[ \alpha = \frac{j - k}{j - i} \]

where \( i \leq h \leq j \), \( h(i) \), and \( h(j) \) are the subchannel estimations of scattered pilots which are located in different OFDM symbols and dependent on SP phase parameter.

The active carriers and corresponding subchannel carriers are output for use in the equalizer model. The inserted zeros in DTV_DVBLoadIFFTBuff are discarded.

References

**DTV_DVBDemuxOFDMSym**

**Description**  
Data and TPS demux for DVB-T symbol

**Library**  
DTV, DVB-T

**Class**  
SDFDTV_DVBDemuxOFDMSym

**Required Licenses**

**Parameters**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Default</th>
<th>Type</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Carriers</td>
<td>number of carriers in one OFDM symbol</td>
<td>1705</td>
<td>int</td>
<td>1705 for 2k mode; 6817 for 8k mode</td>
</tr>
<tr>
<td>Data</td>
<td>number of input data in one OFDM symbol</td>
<td>1512</td>
<td>int</td>
<td>1512 for 2k mode; 6048 for 8k mode</td>
</tr>
<tr>
<td>SPperiod</td>
<td>distance in carriers between nearby scattered pilots</td>
<td>12</td>
<td>int</td>
<td>([0, \infty))</td>
</tr>
<tr>
<td>SPstart</td>
<td>start position of scattered pilots in carriers</td>
<td>0</td>
<td>int</td>
<td>([0, \infty))</td>
</tr>
<tr>
<td>SPOffset</td>
<td>offset value of scattered pilots in each symbol</td>
<td>3</td>
<td>int</td>
<td>([0, \infty))</td>
</tr>
<tr>
<td>SPphase</td>
<td>initial phase of scattered pilots</td>
<td>0</td>
<td>int</td>
<td>([0, \text{SPperiod}/\text{SPOffset}-1] )††</td>
</tr>
</tbody>
</table>

† In DVB-T systems: SPperiod=12, SPstart=0, SPOffset=3

†† SPphase=3 in the OFDM receiver (because DTV_MLEstimator makes one OFDM symbol delay).

**Pin Inputs**

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>input</td>
<td>equalized signals before de-multiplexer</td>
<td>complex</td>
</tr>
</tbody>
</table>
DVB-T Components

Pin Outputs

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>data</td>
<td>OFDM demodulation data</td>
<td>complex</td>
</tr>
<tr>
<td>3</td>
<td>TPS</td>
<td>OFDM demodulation TPS</td>
<td>complex</td>
</tr>
</tbody>
</table>

Notes/Equations

1. The model is used to demultiplex the DVB-T OFDM symbols (such as QPSK, 16-QAM, and 64-QAM modulation) into TSP (transport stream packet) data, TPS (transmission parameter signal) data. Figure 3-3 illustrates the frame structure.

![Figure 3-3. Frame Structure](image)

2. The number of IntState of Length determines the number of CP (continual pilot) and TPS in each symbol; the position of CP and TPS are determined according to Table 3-6 and Table 3-7, respectively.

PRBS sequences in each segment are generated according to Figure 3-4; the initial sets of PRBS register is 1111111111. The PRBS in initialized so that the first output bit from the PRBS coincides with the first active carrier. A new value is generated by the PRBS on each carrier used in each segment (whether or not it is a pilot).

Positions of the corresponding scattered pilots are generated as follows. For the symbol of index \( l \) (ranging from 0 to 67), carriers for which index \( k \) belongs to subset
\[ k = K_{\text{min}} + 3 \times l \mod 4 + 12p \mid p \text{ integer}, \ p \geq 0, \ k \in \{K_{\text{min}}, K_{\text{max}}\} \]

are scattered pilots positions.

where \( p \) is an integer that takes all possible values \( \geq 0 \), provided the resulting value for \( k \) does not exceed the valid range \([K_{\text{min}}, K_{\text{max}}]\); \( K_{\text{min}} = 0 \), and \( K_{\text{max}} = 1704 \) for 2k mode or 6816 for 8k mode. SPperiod, SPstart, SPoffset, and SPphase parameters control the scattered pilots positions.

After determining the CP, TPS, and SP positions in each symbol and the value of the PRBS sequence in all active carriers in the symbol, the model demultiplexes the input data into TPS and TSP data. According to the TPS position and the PRBS sequence, one transmitted TPS bit is determined by the mean value of the TPS positions.

\[
\text{TPS} = \frac{1}{\text{NTPS}} \sum_{l=1}^{\text{NTPS}} x_{\text{TPSposition}[l]} \cdot \text{PilotValue}_{\text{TPSposition}[l]}
\]

where NTPS is the number of TPS in one symbol; PilotValue is the PRBS sequence in symbol; \( x[i] \) is the input data.

Except for TPS, CP, and SP positions, the remaining positions in one segment are the TSP data positions.

\[
\text{Data}[i] = x[i]
\]

### Table 3-6. Carrier Indices for Continual Pilot Carriers

<table>
<thead>
<tr>
<th>Continual pilot carrier positions (index number k)</th>
<th>2k mode</th>
<th>8k mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>2k mode</td>
<td>8k mode</td>
<td></td>
</tr>
<tr>
<td>0 48 54 87 141 156 192 201 255 279 282 333 432 450 483 525</td>
<td>0 48 54 87 141 156 192 201 255 279 282 333 432 450 483 525</td>
<td></td>
</tr>
<tr>
<td>531 618 636 714 759 765 780 804 873 888 918 939 942 969 984</td>
<td>531 618 636 714 759 765 780 804 873 888 918 939 942 969 984</td>
<td></td>
</tr>
<tr>
<td>1050 1101 1107 1110 1137 1140 1146 1206 1269 1323 1377 1491 1683 1704</td>
<td>1050 1101 1107 1110 1137 1140 1146 1206 1269 1323 1377 1491 1683 1704</td>
<td></td>
</tr>
</tbody>
</table>

DTV_DVBDemuxOFDMSym 3-19
DVB-T Components

Table 3-7. Carrier Indices for TPS Carriers

<table>
<thead>
<tr>
<th>2k mode</th>
<th>8k mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>34 50 209 346 413 569 595 688 790 901 1073 1219 1262 1286 1469 1594 1687</td>
<td>34 50 209 346 413 569 595 688 790 901 1073 1219 1262 1286 1469 1594 1687 1738 1754 1913 2050 2273 2299 2392 2494 2605 2777 2923 2966 2990 3173 3298 3391 3442 3458 3617 3754 3821 3977 4003 4096 4198 4309 4481 4627 4670</td>
</tr>
</tbody>
</table>

Figure 3-4. PRBS Sequence Generation

References

**DTV_DVBLoadIFFTBuff**

**Description**  Data stream loader into IFFT buffer for DVB-T

**Library**  DTV, DVB-T

**Class**  SDFDTV_DVBLoadIFFTBuff

**Parameters**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Default</th>
<th>Type</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Carriers</td>
<td>number of carriers in one OFDM symbol</td>
<td>1705</td>
<td>int</td>
<td>1705 for 2k mode; 6817 for 8k mode</td>
</tr>
<tr>
<td>Order</td>
<td>IFFT points=$2^\text{Order}$</td>
<td>13</td>
<td>int</td>
<td>$[1, \infty]$†</td>
</tr>
</tbody>
</table>

† Order is the order of IFFT; it must satisfy $2^\text{Order} \geq \text{Carriers}$

**Pin Inputs**

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>input</td>
<td>transmitted signal before IFFT</td>
<td>complex</td>
</tr>
</tbody>
</table>

**Pin Outputs**

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>output</td>
<td>IFFT input signal, zero padded</td>
<td>complex</td>
</tr>
</tbody>
</table>

**Notes/Equations**

1. The model is used to load the transmission data into the IFFT buffer, which is used only in DVB-T.

2. Implementation

   Assume $x(0), x(1), \ldots, x(N-1)$ are the changed signal segments,
DVB-T Components

where

\[ N = \text{Length} \times \text{Segments} + 1 \]

\[ M = 2^{\text{Order}} \]

\[ y(0), y(1), \ldots, y(M-1) \] are the model outputs.

Data loading is performed as follows:

\[ y(i) = x\left(\frac{N}{2} + i\right) \quad i = 0, \ldots, \frac{N + 1}{2} - 1 \]

\[ y(i) = 0 \quad i = \frac{N + 1}{2}, \ldots, M - \frac{N}{2} - 1 \]

\[ y(i) = x\left(i - M + \frac{N}{2}\right) \quad i = M - \frac{N}{2}, \ldots, M \]

References

**DTV_DVBMuxOFDMSym**

**Description**  OFDM symbol multiplexer for DVB-T

**Library**  DTV, DVB-T

**Class**  SDFDTV_DVBMuxOFDMSym

**Parameters**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Default</th>
<th>Type</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Carriers</td>
<td>number of carriers in one OFDM symbol</td>
<td>1705</td>
<td>int</td>
<td>1705 for 2k mode; 6817 for 8k mode</td>
</tr>
<tr>
<td>Data</td>
<td>number of input data in one OFDM symbol</td>
<td>1512</td>
<td>int</td>
<td>1512 for 2k mode; 6048 for 8k mode</td>
</tr>
<tr>
<td>SPperiod</td>
<td>distance in carriers between nearby scattered pilots</td>
<td>12</td>
<td>int</td>
<td>[0, ∞)†</td>
</tr>
<tr>
<td>SPstart</td>
<td>start position of scattered pilots in carriers</td>
<td>0</td>
<td>int</td>
<td>(0, ∞)†</td>
</tr>
<tr>
<td>SPOffset</td>
<td>offset value of SPstart in each symbol</td>
<td>3</td>
<td>int</td>
<td>(0, ∞)†</td>
</tr>
</tbody>
</table>

† In DVB-T systems: SPperiod=12, SPstart=0, SPOffset=3

**Pin Inputs**

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>data</td>
<td>data input</td>
<td>complex</td>
</tr>
<tr>
<td>2</td>
<td>TPS</td>
<td>TPS input</td>
<td>complex</td>
</tr>
</tbody>
</table>

**Pin Outputs**

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>output</td>
<td>OFDM symbol data output</td>
<td>complex</td>
</tr>
</tbody>
</table>
Notes/Equations

1. The model is used to multiplex TSP (transport stream packet) and TPS (transmission parameter signal) data into the DVB-T OFDM symbol. Figure 3-5 shows the frame structure.

![Figure 3-5. Frame Structure](image)

TPS pilots and continual pilots between \(K_{\text{ini}}\) and \(K_{\text{mix}}\) are not indicated.

- • boosted pilot
- ○ data

2. The number of IntState of Length determines the number of CP (continual pilot) and TPS in each symbol; the position of CP and TPS are determined according to Table 3-8 and Table 3-9, respectively.

PRBS sequences in each segment are generated according to Figure 3-6; the initial sets of PRBS register is 11111111111. The PRBS is initialized so that the first output bit from the PRBS coincides with the first active carrier. A new value is generated by the PRBS on each carrier in each segment (whether or not it is a pilot).

Positions of the corresponding scattered pilots are generated as follows. For the symbol of index \(l\) (ranging from 0 to 67), carriers for which index \(k\) belongs to subset

\[
\{ k = K_{\text{min}} + 3 \times l \mod 4 + 12 p \mid p \text{ integer, } p \geq 0, k \in [K_{\text{min}}, K_{\text{mix}}]\}
\]

are scattered pilot positions.

where \(p\) is an integer that takes all possible values \(\geq 0\), provided the resulting value for \(k\) does not exceed the valid range \([K_{\text{min}}, K_{\text{mix}}]; K_{\text{min}}=0, and [\]

---

3-24  DTV_DVBmuxOFDMSym
$K_{\text{max}} = 1704$ for 2k mode or 6816 for 8k mode. SP period, SP start, and SP offset parameters control the scattered pilots positions.

After determining the CP, TPS, and SP positions in each symbol and the value of the PRBS sequence in all active carriers in one symbol, the model demultiplexes the input data into TPS and TSP data. According to the TPS position and the PRBS sequence, one TPS bit is output in each TPS position.

$$x[\text{TPS position}[l]] = \text{PilotValue}[\text{TPS position}[l]] \times \text{TPS}$$

where $N_{\text{TPS}}$ is the number of TPS in one symbol; PilotValue is the PRBS sequence in symbol; $x[i]$ is the OFDM symbol data.

Except for TPS, CP, and SP positions, the remaining positions in one segment are the TSP data positions.

$$x[i] = \text{Data}[i]$$

### Table 3-8. Carrier Indices for Continual Pilot Carriers

<table>
<thead>
<tr>
<th>Continual pilot carrier positions (index number $k$)</th>
<th>2k mode</th>
<th>8k mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 48 54 87 141 156 192 201 255 279 282 333 432 450 483 525</td>
<td>0 48 54 87 141 156 192 201 255 279 282 333 432 450 483 525</td>
<td>531 618 636 714 759 765 780 804 873 888 918 939 942 969 984</td>
</tr>
<tr>
<td>0 48 54 87 141 156 192 201 255 279 282 333 432 450 483 525</td>
<td>531 618 636 714 759 765 780 804 873 888 918 939 942 969 984</td>
<td>1050 1101 1107 1110 1137 1140 1146 1206 1269 1323 1377 1491</td>
</tr>
<tr>
<td>1683 1704</td>
<td>1683 1704</td>
<td>1683 1704</td>
</tr>
<tr>
<td>531 618 636 714 759 765 780 804 873 888 918 939 942 969 984</td>
<td>2037 2136 2154 2187 2229 2232 2322 2340 2418 2463 2469 2484</td>
<td>2508 2577 2592 2622 2646 2673 2688 2754 2805 2811 2814</td>
</tr>
<tr>
<td>1050 1101 1107 1110 1137 1140 1146 1206 1269 1323 1377 1491</td>
<td>2841 2644 2650 2910 2973 3027 3081 3195 3387 3408 3456 3462</td>
<td>3495 3549 3594 3600 3609 3663 3687 3690 3741 3840 3858 3891</td>
</tr>
<tr>
<td>1683 1704</td>
<td>3933 3939 4026 4044 4122 4167 4173 4188 4212 4281 4296 4326</td>
<td>3933 3939 4026 4044 4122 4167 4173 4188 4212 4281 4296 4326</td>
</tr>
<tr>
<td>531 618 636 714 759 765 780 804 873 888 918 939 942 969 984</td>
<td>4347 4350 4377 4392 4456 4509 4515 4518 4545 4548 4554 4561</td>
<td>4677 4731 4785 4899 5091 5112 5160 5166 5199 5253 5268 5304</td>
</tr>
<tr>
<td>1050 1101 1107 1110 1137 1140 1146 1206 1269 1323 1377 1491</td>
<td>5313 5367 5391 5394 5445 5544 5562 5595 5637 5643 5730 5748</td>
<td>5826 5871 5877 5892 5916 5985 6000 6030 6051 6054 6081 6096</td>
</tr>
<tr>
<td>1683 1704</td>
<td>5826 5871 5877 5892 5916 5985 6000 6030 6051 6054 6081 6096</td>
<td>6162 6213 6219 6222 6249 6252 6258 6318 6381 6435 6489 6603</td>
</tr>
<tr>
<td>531 618 636 714 759 765 780 804 873 888 918 939 942 969 984</td>
<td>6795 6816</td>
<td>6795 6816</td>
</tr>
</tbody>
</table>

### Table 3-9. Carrier Indices for TPS Carriers

<table>
<thead>
<tr>
<th>2k mode</th>
<th>8k mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>34 50 209 346 413 569 595 668 790 901 1073 1219 1262 1286 1469 1594 1687</td>
<td>34 50 209 346 413 569 595 668 790 901 1073 1219 1262 1286 1469 1594 1687</td>
</tr>
<tr>
<td>1738 1754 1913 2050 2117 2273 2299 2392 2494 2605 2777 2923 2966 2990 3173</td>
<td>3298 3391 3442 3458 3617 3754 3821 3977 4003 4098 4198 4309 4481 4627 4670</td>
</tr>
<tr>
<td>3469 4677 5002 5095 5146 5162 5321 5458 5525 5681 5707 5800 5902 6013 6185</td>
<td>6331 6374 6398 6581 6706 6799</td>
</tr>
<tr>
<td>6331 6374 6398 6581 6706 6799</td>
<td></td>
</tr>
</tbody>
</table>
DVB-T Components

Figure 3-6. PRBS Sequence Generation

Initialization sequence

PRBS sequence: 1111111111100 ...

References

DTV_DVBSymDeinterlv2b

Description    Symbol de-interleaver with 2 branches
Library        DTV, DVB-T
Class          SDFDTV_DVBSymDeinterlv2b
Derived From   DTV_DVBSymInterlv

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Default</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mode</td>
<td>DVB OFDM mode: DVB 2k mode, DVB 8k mode</td>
<td>DVB 2k mode</td>
<td>enum</td>
</tr>
</tbody>
</table>

Pin Inputs

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>input</td>
<td>interleaved OFDM symbol bits</td>
<td>anytype</td>
</tr>
</tbody>
</table>

Pin Outputs

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>outb0</td>
<td>data symbol bit 0</td>
<td>anytype</td>
</tr>
<tr>
<td>3</td>
<td>outb1</td>
<td>data symbol bit 1</td>
<td>anytype</td>
</tr>
</tbody>
</table>

Notes/Equations

1. This model is used to perform symbol de-interleaving in DVB-T systems. Each firing, it consumes 1512 (2k mode) or 6048 (8k mode) data symbols at the input and outputs them to each output pin after de-interleaving.

2. This model is the reverse of the process used for DTV_DVBSymInterlv2b. The interleaving algorithm is described in DTV_DVBSymInterlv2b.

3. This model is one of three symbol de-interleavers with data symbol bit output. 2-bit, 4-bit and 6-bit data symbol de-interleavers are used for QPSK, 16-QAM,
DVB-T Components

and 64-QAM, respectively, after constellation mapping de-mapping, a complex symbol interleaver is used before constellation de-mapping. These models use the same permutation sequence.

References

**DTV_DVBSymDeinterlv4b**

**Description**  Symbol de-interleaver with 4 branches

**Library**  DTV, DVB-T

**Class**  SDFDTV_DVBSymDeinterlv4b

**Derived From**  DTV_DVBSymInterlv

**Parameters**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Default</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mode</td>
<td>DVB OFDM mode: DVB 2k mode, DVB 8k mode</td>
<td>DVB 2k mode</td>
<td>enum</td>
</tr>
</tbody>
</table>

**Pin Inputs**

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>input</td>
<td>interleaved OFDM symbol bits</td>
<td>anytype</td>
</tr>
</tbody>
</table>

**Pin Outputs**

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>outb0</td>
<td>data symbol bit 0</td>
<td>anytype</td>
</tr>
<tr>
<td>3</td>
<td>outb1</td>
<td>data symbol bit 1</td>
<td>anytype</td>
</tr>
<tr>
<td>4</td>
<td>outb2</td>
<td>data symbol bit 2</td>
<td>anytype</td>
</tr>
<tr>
<td>5</td>
<td>outb3</td>
<td>data symbol bit 3</td>
<td>anytype</td>
</tr>
</tbody>
</table>

**Notes/Equations**

1. This model is used to perform symbol de-interleaving in DVB-T systems. Each firing, it consumes 1512 (2k mode) or 6048 (8k mode) data symbols at the input and outputs them to each output pin after de-interleaving.

2. This model is the reverse of the process used for DTV_DVBSymInterlv4b. The interleaving algorithm is described in DTV_DVBSymInterlv4b.
3. This model is one of three symbol de-interleavers with data symbol bit output. 2-bit, 4-bit and 6-bit data symbol de-interleavers are used for QPSK, 16-QAM, and 64-QAM, respectively, after constellation mapping de-mapping, a complex symbol interleaver is used before constellation de-mapping. These models use the same permutation sequence.

References

DVT_DVBSymDeinterlv6b

Description  Symbol de-interleaver with 6 branches
Library  DTV, DVB-T
Class  SDFDTV_DVBSymDeinterlv6b
Derived From  DTV_DVBSymInterlv

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Default</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mode</td>
<td>DVB OFDM mode: DVB 2k mode, DVB 8k mode</td>
<td>DVB 2k mode</td>
<td>enum</td>
</tr>
</tbody>
</table>

Pin Inputs

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>input</td>
<td>interleaved OFDM symbol bits</td>
<td>anytype</td>
</tr>
</tbody>
</table>

Pin Outputs

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>outb0</td>
<td>data symbol bit 0</td>
<td>anytype</td>
</tr>
<tr>
<td>3</td>
<td>outb1</td>
<td>data symbol bit 1</td>
<td>anytype</td>
</tr>
<tr>
<td>4</td>
<td>outb2</td>
<td>data symbol bit 2</td>
<td>anytype</td>
</tr>
<tr>
<td>5</td>
<td>outb3</td>
<td>data symbol bit 3</td>
<td>anytype</td>
</tr>
<tr>
<td>6</td>
<td>outb4</td>
<td>data symbol bit 4</td>
<td>anytype</td>
</tr>
<tr>
<td>7</td>
<td>outb5</td>
<td>data symbol bit 5</td>
<td>anytype</td>
</tr>
</tbody>
</table>

Notes/Equations

1. This model is used to perform symbol de-interleaving in DVB-T systems. Each firing, it consumes 1512 (2k mode) or 6048 (8k mode) data symbols at the input and outputs them to each output pin after de-interleaving.
DVB-T Components

2. This model is the reverse of the process used for DTV_DVBSymInterlv6b. The interleaving algorithm is described in DTV_DVBSymInterlv6b.

3. This model is one of three symbol de-interleavers with data symbol bit output. 2-bit, 4-bit and 6-bit data symbol de-interleavers are used for QPSK, 16-QAM, and 64-QAM, respectively, after constellation mapping de-mapping, a complex symbol interleaver is used before constellation de-mapping. These models use the same permutation sequence.

References

DTV_DVBSymInterlv2b

Description   Symbol interleaver with 2 branches
Library       DTV, DVB-T
Class         SDFDTV_DVBSymInterlv2b
Derived From  DTV_DVBSymInterlv

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Default</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mode</td>
<td>DVB OFDM mode: DVB 2k mode, DVB 8k mode</td>
<td>DVB 2k mode</td>
<td>enum</td>
</tr>
</tbody>
</table>

Pin Inputs

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>inb0</td>
<td>data symbol bit 0</td>
<td>anytype</td>
</tr>
<tr>
<td>2</td>
<td>inb1</td>
<td>data symbol bit 1</td>
<td>anytype</td>
</tr>
</tbody>
</table>

Pin Outputs

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>output</td>
<td>interleaved OFDM symbol bits</td>
<td>anytype</td>
</tr>
</tbody>
</table>

Notes/Equations

1. This model is used to perform symbol interleaving in DVB-T systems. Each firing, it consumes 1512 (2k mode) or 6048 (8k mode) bits data in each input pin, and outputs them after interleaving. The interleaving algorithm is described in note 2.

2. This model is one of the three symbol interleavers with data symbol bit inputs. 2-bit, 4-bit, and 6-bit data symbol interleavers are used for QPSK, 16-QAM, and 64-QAM, respectively, before constellation mapping; a complex symbol
interleaver is used after constellation mapping. These models use the same permutation sequence.

Interleaving Algorithm
Define input vector \( Y' \) and interleaved vector \( Y \) as:

\[
Y' = (y'_0, y'_1, y'_2, \ldots, y'_{N_{\text{max}}-1})
\]
\[
Y = (y_0, y_1, y_2, \ldots, y_{N_{\text{max}}-1})
\]

\[Y_{H(q)} = Y' q \text{ for even symbols for } q = 0, \ldots, N_{\text{max}} - 1\]

\[Y_q = Y'_{H(q)} \text{ for odd symbols for } q = 0, \ldots, N_{\text{max}} - 1\]

where \( N_{\text{max}} = 1512 \) in 2k mode and \( N_{\text{max}} = 6048 \) in 8k mode.

\( H(q) \) is a permutation function defined by the following:

An \( (N_r - 1) \) bit binary word \( R'_i \) is defined, with
\[
N_r = \log_2(M_{\text{max}}),
\]
where \( M_{\text{max}} = 1512 \) in 2k mode and \( M_{\text{max}} = 2048 \) in the 8k mode,
where \( R'_i \) takes the following values:

\[i = 0,1: \ R'_i[N_r - 2, N_r - 3, \ldots, 1, 0] = (0, 0, \ldots, 0, 0)\]

\[i = 2: \ R'_i[N_r - 2, N_r - 3, \ldots, 1, 0] = (0, 0, \ldots, 0, 1)\]

\[2 < i < M_{\text{max}}: \]

\[\{R'_i[N_r - 3, N_r - 4, \ldots, 1, 0] = R'_{i-1}[N_r - 2, N_r - 3, \ldots, 1, \xi\}
\]

in the 2k mode: \( R'_i[9] = (R'_{i-1}[0] \oplus R'_{i-1}[3])\)

in the 8k mode: \( R'_i[9] = (R'_{i-1}[0] \oplus R'_{i-1}[1] \oplus R'_{i-1}[4] \oplus R'_{i-1}[6])\)

A vector \( R'_i \) is derived from vector \( R'_i \) by bit permutations given in Table 3-10 and Table 3-11.
The permutation function \( H(q) \) is defined by the algorithm:

\[
\text{q = 0; for (i = 0; i < M_{\text{max}}; i = i+1)} \\
\{ \\
\quad H(q) < N_{\text{max}} \quad \text{q = q+1;}
\}
\]

The algorithm used to generate the permutation function is illustrated in Figure 3-7 for the 2k mode and Figure 3-8 for the 8k mode.

Each input signal generates its own input \( Y' \) vector and interleaved \( Y \) vector. All interleaved \( Y \) vectors are synthesized into the output signal.
DVB-T Components

Figure 3-7. 2k Mode Symbol Interleaver Address Generation

Figure 3-8. 8k Mode Symbol Interleaver Address Generation

References

DTV_DVBSymInterlv4b

Description  Symbol interleaver with 4 branches
Library     DTV, DVB-T
Class       SDFDTV_DVBSymInterlv4b
Derived From  DTV_DVBSymInterlv

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Default</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mode</td>
<td>DVB OFDM mode: DVB 2k mode, DVB 8k mode</td>
<td>DVB 2k mode</td>
<td>enum</td>
</tr>
</tbody>
</table>

Pin Inputs

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>inb0</td>
<td>data symbol bit 0</td>
<td>anytype</td>
</tr>
<tr>
<td>2</td>
<td>inb1</td>
<td>data symbol bit 1</td>
<td>anytype</td>
</tr>
<tr>
<td>3</td>
<td>inb2</td>
<td>data symbol bit 2</td>
<td>anytype</td>
</tr>
<tr>
<td>4</td>
<td>inb3</td>
<td>data symbol bit 3</td>
<td>anytype</td>
</tr>
</tbody>
</table>

Pin Outputs

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>output</td>
<td>interleaved OFDM symbol bits</td>
<td>anytype</td>
</tr>
</tbody>
</table>

Notes/Equations

1. This model is used to perform symbol interleaving in DVB-T systems. Each firing, it consumes 1512 (2k mode) or 6048 (8k mode) bits data in each input pin, and outputs them after interleaving. The interleaving algorithm is the same as that described for DTV_DVBSymInterlv2b except for the input bit streams.
DVB-T Components

2. This model is one of the three symbol interleavers with data symbol bit inputs. 2-bit, 4-bit, and 6-bit data symbol interleavers are used for QPSK, 16-QAM, and 64-QAM, respectively, before constellation mapping; a complex symbol interleaver is used after constellation mapping. These models use the same permutation sequence.

References

DTV_DVBSymInterlv6b

Description  Symbol interleaver with 6 branches
Library       DTV, DVB-T
Class         SDFDTV_DVBSymInterlv6b
Derived From  DTV_DVBSymInterlv

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Default</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mode</td>
<td>DVB OFDM mode: DVB 2k mode, DVB 8k mode</td>
<td>DVB 2k mode</td>
<td>enum</td>
</tr>
</tbody>
</table>

Pin Inputs

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>inb0</td>
<td>data symbol bit 0</td>
<td>anytype</td>
</tr>
<tr>
<td>2</td>
<td>inb1</td>
<td>data symbol bit 1</td>
<td>anytype</td>
</tr>
<tr>
<td>3</td>
<td>inb2</td>
<td>data symbol bit 2</td>
<td>anytype</td>
</tr>
<tr>
<td>4</td>
<td>inb3</td>
<td>data symbol bit 3</td>
<td>anytype</td>
</tr>
<tr>
<td>5</td>
<td>inb4</td>
<td>data symbol bit 4</td>
<td>anytype</td>
</tr>
<tr>
<td>6</td>
<td>inb5</td>
<td>data symbol bit 5</td>
<td>anytype</td>
</tr>
</tbody>
</table>

Pin Outputs

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>output</td>
<td>interleaved OFDM symbol bits</td>
<td>anytype</td>
</tr>
</tbody>
</table>

Notes/Equations

1. This model is used to perform symbol interleaving in DVB-T systems. Each firing, it consumes 1512 (2k mode) or 6048 (8k mode) bits data in each input pin.
DVB-T Components

and outputs them after interleaving. The interleaving algorithm is the same as that described for DTV_DVBSymInterlv2b except for the input bit streams.

2. This model is one of the three symbol interleavers with data symbol bit inputs. 2-bit, 4-bit, and 6-bit data symbol interleavers are used for QPSK, 16-QAM, and 64-QAM, respectively, before constellation mapping; a complex symbol interleaver is used after constellation mapping. These models use the same permutation sequence.

References

**DTV_DVBSymInterlvCx**

**Description**  Symbol interleaver  
**Library**  DTV, DVB-T  
**Class**  SDFDTV_DVBSymInterlvCx  
**Derived From**  DTV_DVBSymInterlv

**Parameters**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Default</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mode</td>
<td>DVB OFDM mode: DVB 2k mode, DVB 8k mode</td>
<td>DVB 2k mode</td>
<td>enum</td>
</tr>
<tr>
<td>Option</td>
<td>operation option: Interleaving, De-interleaving</td>
<td>Interleaving</td>
<td>enum</td>
</tr>
</tbody>
</table>

**Pin Inputs**

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>input</td>
<td>OFDM symbols after constellation mapping</td>
<td>anytype</td>
</tr>
</tbody>
</table>

**Pin Outputs**

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>output</td>
<td>interleaved OFDM symbols</td>
<td>anytype</td>
</tr>
</tbody>
</table>

**Notes/Equations**

1. This model is used to perform complex symbol interleaving in DVB-T systems. Each firing, it consumes 1512 (2k mode) or 6048 (8k mode) data bits at the input and outputs them after interleaving. The interleaving algorithm is the same as that described for DTV_DVBSymInterlv2b.

2. Besides this complex symbol interleaver, there are three symbol interleavers using data bits as input. The data symbol interleavers are used before...
DVB-T Components

constellation mapping; the complex symbol interleaver is used after constellation mapping. These models use the same permutation sequence.

References

**DTV_DVBTHierInnerDeInterlv**

**Description**  
Inner de-interleaver for 16QAM, 64QAM hierarchical transmission modes

**Library**  
DTV, DVB-T

**Required Licenses**

**Parameters**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Default</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mode</td>
<td>transmission mode: DVB 2k mode, DVB 8k mode</td>
<td>DVB 2k mode</td>
<td>enum</td>
</tr>
<tr>
<td>MappingMode</td>
<td>signal constellations and mapping: QAM-16, QAM-64</td>
<td>QAM-16</td>
<td>enum</td>
</tr>
</tbody>
</table>

**Pin Inputs**

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Symbols</td>
<td>input of inner de-interleaver</td>
<td>anytype</td>
</tr>
</tbody>
</table>

**Pin Outputs**

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>HP_Bits</td>
<td>The high priority bit stream for hierarchical modes</td>
<td>anytype</td>
</tr>
<tr>
<td>3</td>
<td>LP_Bitstream</td>
<td>The low priority bit stream for hierarchical modes</td>
<td>anytype</td>
</tr>
</tbody>
</table>

**Notes/Equations**

1. This subnetwork model is used to perform inner de-interleaving for 16QAM and 64QAM for hierarchical transmission. 16QAM and 64QAM functions are performed by an if-else structure. The schematic for this subnetwork is shown in Figure 3-9.
DVB-T Components

Figure 3-9. DTV_DVBTHierInnerDeinterlv Subnetwork

2. The Mode parameter specifies the 2k or 8k mode.
3. The MappingMode parameter specifies QAM-16 or QAM-64.

References

**DTV_DVBTHierInnerInterlv**

**Description**  Inner interleaver for 16QAM, 64QAM hierarchical transmission mode

**Library**  DTV, DVB-T

**Required Licenses**

**Parameters**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Default</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mode</td>
<td>transmission mode: DVB 2k mode, DVB 8k mode</td>
<td>DVB 2k mode</td>
<td>enum</td>
</tr>
<tr>
<td>MappingMode</td>
<td>signal constellations and mapping: QAM-16, QAM-64</td>
<td>QAM-16</td>
<td>enum</td>
</tr>
</tbody>
</table>

**Pin Inputs**

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>HP_Bits</td>
<td>The high priority bit stream for hierarchical modes</td>
<td>anytype</td>
</tr>
<tr>
<td>2</td>
<td>LP_Bits</td>
<td>The low priority bit stream for hierarchical modes</td>
<td>anytype</td>
</tr>
</tbody>
</table>

**Pin Outputs**

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>Symbols</td>
<td>output of inner interleaver</td>
<td>anytype</td>
</tr>
</tbody>
</table>

**Notes/Equations**

1. This subnetwork model is used to perform 16QAM and 64QAM inner interleaving for hierarchical transmission. 16QAM and 64QAM inner interleaver is integrated; functions are performed by an if-else structure. The schematic for this subnetwork is shown in Figure 3-10.
2. The Mode parameter specifies the 2k or 8k mode.
3. The MappingMode parameter specifies QAM-16 or QAM-64.

References

**Description**  
DVB-T hierarchical receiver

**Library**  
DTV, DVB-T

**Required Licenses**

**Parameters**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Default</th>
<th>Sym</th>
<th>Type</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mode</td>
<td>transmission mode: DVB 2k mode, DVB 8k mode</td>
<td>DVB 2k mode</td>
<td>enum</td>
<td></td>
<td></td>
</tr>
<tr>
<td>OversamplingOption</td>
<td>oversampling ratio option: Ratio 1, Ratio 2, Ratio 4, Ratio 8, Ratio 16, Ratio 32</td>
<td>Ratio 1</td>
<td>S</td>
<td>enum</td>
<td></td>
</tr>
<tr>
<td>GuardInterval</td>
<td>guard interval (frac FFT size)</td>
<td>0.125</td>
<td>real</td>
<td>[0, 1]</td>
<td></td>
</tr>
<tr>
<td>CodeRateHP</td>
<td>high priority stream current code rate: HP 1/2, HP 2/3, HP 3/4, HP 5/6, HP 7/8</td>
<td>HP 1/2</td>
<td>enum</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CodeRateLP</td>
<td>low priority stream current code rate: LP 1/2, LP 2/3, LP 3/4, LP 5/6, LP 7/8</td>
<td>LP 1/2</td>
<td>enum</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MappingMode</td>
<td>signal constellations and mapping: QAM-16, QAM-64</td>
<td>QAM-16</td>
<td>enum</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Alpha</td>
<td>non-uniform factor for DVB-T.</td>
<td>1</td>
<td>int</td>
<td>[1, (\infty)]</td>
<td></td>
</tr>
<tr>
<td>SoftDecision</td>
<td>soft decision viterbi decoding type: Soft, Hard, Cochannel</td>
<td>Soft</td>
<td>enum</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TrunLen</td>
<td>path memory truncation length of Viterbi decoding algorithm, in bytes</td>
<td>10</td>
<td>int</td>
<td>[5, (\infty)]</td>
<td></td>
</tr>
</tbody>
</table>

**Pin Inputs**

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Rx_signal</td>
<td>received OFDM signal to be demodulated</td>
<td>complex</td>
</tr>
</tbody>
</table>
DVB-T Components

Pin Outputs

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>HPBytes_Decoded</td>
<td>decoded bytes after Reed Solomon decoder with high priority</td>
<td>int</td>
</tr>
<tr>
<td>3</td>
<td>LPBytes_Decoded</td>
<td>decoded bytes after Reed Solomon decoder with low priority</td>
<td>int</td>
</tr>
<tr>
<td>4</td>
<td>HPBytes_AfterViterbi</td>
<td>decoded bytes after Viterbi decoder with high priority</td>
<td>int</td>
</tr>
<tr>
<td>5</td>
<td>LPBytes_AfterViterbi</td>
<td>decoded bytes after Viterbi decoder with low priority</td>
<td>int</td>
</tr>
<tr>
<td>6</td>
<td>TPS_Bits</td>
<td>decoded TPS information bits</td>
<td>int</td>
</tr>
<tr>
<td>7</td>
<td>constellation</td>
<td>constellation signal after OFDM demodulation</td>
<td>complex</td>
</tr>
</tbody>
</table>

Notes/Equations

1. This subnetwork model implements a DVB-T hierarchical receiver. The schematic for this subnetwork is shown in Figure 3-11.

Figure 3-11. DTV_DVBTHierReceiver Schematic
2. Output Delays

The constellation pin has a delay of 8 OFDM symbols. The $8 \times 1705$ (2k mode) or $8 \times 6817$ (8k mode) constellation output signals are not used for EVM calculation.

DTV_BCHDecoder uses a 68-bit block; so, a Delay component with $N=60$ is inserted before DTV_DVBTPSDemod.

Pin TPS_Bits has one DVB-T frame (68 OFDM symbols) delay; the first 53 bits at TPS_Bits are not used.

LPBytes_AfterViterbi and HPBytes_AfterViterbi each have 8 OFDM symbol delays.

The inner decoding (or convolutional decoding) is based on MPEG-2 transport MUX packet. So, the LPBytes_Decoded, HPBytes_Decoded, HPBytes_AfterViterbi, and LPBytes_AfterViterbi delays must be multiples of coded (204) and uncoded (188) transport MUX packet blocks.

The information bytes per one OFDM symbol in HPBytes_AfterViterbi is calculated as follows:

\[
\text{DataPerOFDMSymbol} = 1512 \text{ (2k mode) or 6048 (8k mode)}
\]

\[
\text{CodedBlockSize} = 204
\]

\[
\text{HPCodeRate} = 1/2, 2/3, 3/4, 5/6 \text{ or } 7/8
\]

\[
\text{DelayOFDMSymbols} = 8
\]

\[
\text{HPInfoBytes} = \text{DelayOFDMSymbols} \times \text{DataPerOFDMSymbol} \times \text{HPCodeRate}/8
\]

The block delay of HPBytes_AfterViterbi is $\text{Delay_HPAfterViterbi}$ and calculated as follows:

\[
\text{HPDelayBytes} = \text{DelayOFDMSymbols} \times \text{HPInfoBytes} - (\text{int}(\text{DelayOFDMSymbols} \times \text{HPInfoBytes}/\text{CodedBlockSize})) \times \text{CodedBlockSize}
\]

\[
N_\text{HP} = \text{int}((\text{TrunLen} + \text{LPDelayBytes})/\text{CodedBlockSize})
\]

\[
\text{Delay_HPAfterViterbi} = 1 + \text{int}((\text{DelayOFDMSymbols} \times \text{HPInfoBytes})/\text{CodedBlockSize}) + N_\text{HP}
\]

Block size is 204 bytes.

The information bytes per one OFDM symbol in LPBytes_AfterViterbi is calculated as follows:
DVB-T Components

DataPerOFDMSymbol = 1512 (2k mode) or 6048 (8k mode)
CodedBlockSize = 204
BitsPerQAMmappingLP = 2 for 16-QAM or 4 for 64-QAM,
LPCodeRate = 1/2, 2/3, 3/4, 5/6 or 7/8
LPInfoBytes = BitsPerQAMmappingLP \times \text{DataPerOFDMSymbol} \times \text{LPCodeRate}/8

The block delay of LPBytes_AfterViterbi is \( \text{Delay\_LPAfterViterbi} \) is calculated as follows:

\[
\text{LPDelayBytes} = \text{DelayOFDMSymbols} \times \text{LPInfoBytes} - (\text{int}(\text{DelayOFDMSymbols} \times \text{LPInfoBytes}/\text{CodedBlockSize})) \times \text{CodedBlockSize},
\]

\[
N_{\_LP} = \text{int}((\text{TrunLen} + \text{LPDelayBytes})/\text{CodedBlockSize}),
\]

\[
\text{Delay\_LPAfterViterbi} = 1 + \text{int}((\text{DelayOFDMSymbols} \times \text{LPInfoBytes})/\text{CodedBlockSize}) + N_{\_LP}.
\]

Block size is 204 bytes.

HPBytes_Decoded and LPBytes_Decoded each have 8 OFDM symbols and 11 transport MUX packet delays.

The block delay of HPBytes_Decoded is:

\[
\text{Delay\_HPDecoded} = 11 + 1 + \text{int}((\text{DelayOFDMSymbols} \times \text{HPInfoBytes})/\text{CodedBlockSize}) + N_{\_HP}.
\]

Block size is 188 bytes.

The block delay of LPBytes_Decoded is

\[
\text{Delay\_LPDecoded} = 11 + 1 + \text{int}((\text{DelayOFDMSymbols} \times \text{LPInfoBytes})/\text{CodedBlockSize}) + N_{\_LP}.
\]

Block size is 188 bytes.

3. Receiver functions are implemented according to the DVB-T Standard.

Start of OFDM symbol is detected by the DTV_DVBT_OFDMDemod subnetwork (shown in Figure 3-12).
DTV_MLEstimator calculates the correlation based on guard interval; DTV_LoadFFTBuff selects the index with the maximum correlation value as the start of FFT.

DTV_DVBCh2DEstimater estimates the complex channel impulse responses for continual and scattered pilot subcarriers. The channel impulse responses of the data subcarriers are 2D interpolated based on the estimated CIR.

Each subcarrier value is divided with a complex estimated channel response coefficient (phase tracking, phase synchronization, and equalization). This simple one-tap frequency domain channel response compensation is implemented by DTV_OFDMEqualizer.

After equalization, DTV_DVBDemuxOFDMSym demultiplexes 1705 or 6817 subcarriers into 1512 (2k mode) or 6048 (8k mode) data subcarriers and 1 demodulated TPS.

The demodulated 1512 or 6048 data subcarriers (such as 16QAM, and 64QAM modulation) are demapped by DTV_Demapper.
DVB-T Components

The demodulated bits are deinterleaved by DTV_DVBTHierInnerDeinterlv and divided into high- and low-priority demodulated bits. The demodulated bits are inner decoded (DTV_InnerDecoder), outer de-interleaved (DTV_InterlvInt), outer decoded (DTV_RSDecoder) and descrambled (DTV_ScrambleByte). The fully-decoded high- and low-priority stream bytes are output as HPBytes_Decoded and LPBytes_Decoded.

After inner-decoding, the partially-decoded high- and low-priority stream bytes are output as HPBytes_AfterViterbi and LPBytes_AfterViterbi.

The TPS signal is demodulated by DTV_DVBTPSDemod; the demodulated bits are decoded by DTV_BCHDecoder.

Output pins of this DTV_DVBTHierReceiver subnetwork correspond to those of DTV_DVBTHierSignalSrc.

This hierarchical receiver works per one OFDM symbol. The number of samples of each OFDM symbol (Rx_signal input pin) is:

- \((1+\text{GuardInterval}) \times 2048 \times \text{OversamplingRatio}\) for 2k mode
- \((1+\text{GuardInterval}) \times 8192 \times \text{OversamplingRatio}\) for 8k mode

The number of signals at the constellation output pin is 1705 (2k mode) or 6817 (8k mode) per one OFDM symbol.
4. Parameter Details

Mode specifies the 2k or 8k transmission mode as defined in DVB-T.

OversamplingOption specifies the transmission signal oversampling ratio 1, 2, 4, 8, 16, 32. If OversamplingOption=Ratio 2, the IFFT size is 4096 for 2k mode or 16384 for 8k mode.

GuardInterval specifies the guard interval (also called cyclic extension) length for each symbol time, as a fraction of the FFT time period. Guard interval types 1/4, 1/8, 1/16 and 1/32 are defined in the DVB-T specification. GuardInterval can be set to any value between 0.0 and 1.0 (not just 1/4, 1/8, 1/16 and 1/32). For proper demodulation, the value must match the guard interval length actually used in the input signal.

CodeRateHP specifies the high-priority stream code rate: 1/2, 2/3, 3/4, 5/6 or 7/8.

CodeRateLP specifies the low-priority stream code rate: 1/2, 2/3, 3/4, 5/6 or 7/8.

MappingMode specifies signal constellation and mapping modes: 16-QAM or 64-QAM.

Alpha is used for signal constellation and mapping as defined in DVB-T specifications. If Alpha=1, 16-QAM or 64-QAM uniform mapping is performed; if Alpha=2 or 4, 16-QAM or 64-QAM in non-uniform mapping is performed.

SoftDecision specifies the Viterbi decoding algorithm mode.

- If SoftDecision=Soft, the Viterbi decoding algorithm is a soft decision decoder and uses channel status information.
- If SoftDecision=Hard, the Viterbi decoding algorithm is a hard decision decoder and does not use channel status information.
- If SoftDecision=Cochannel, the Viterbi decoding algorithm is for co-channel measurement with PAL analog TV and digital DVB-T.

TrunLen sets the truncation length (in bytes) in the Viterbi decoding algorithm. The path memory is 8×TrunLen bits in Viterbi decoding algorithm.

References

DVB-T Components

# DTV_DVBTierReceiver_RF

**Description**: DVB-T RF hierarchical receiver

**Library**: DTV, DVB-T

## Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Default</th>
<th>Sym</th>
<th>Unit</th>
<th>Type</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>RIn</td>
<td>input resistance</td>
<td>DefaultRIn</td>
<td>Ohm</td>
<td>real</td>
<td>(0, ∞)</td>
<td></td>
</tr>
<tr>
<td>ROut</td>
<td>output resistance</td>
<td>DefaultROut</td>
<td>Ohm</td>
<td>real</td>
<td>(0, ∞)</td>
<td></td>
</tr>
<tr>
<td>RTemp</td>
<td>physical temperature, in degrees C</td>
<td>DefaultRTemp</td>
<td>Celsius</td>
<td>real</td>
<td>(-273.15, ∞)</td>
<td></td>
</tr>
<tr>
<td>GainImbalance</td>
<td>Gain imbalance, Q vs I</td>
<td>0.0</td>
<td>dB</td>
<td>real</td>
<td>(-∞, ∞)</td>
<td></td>
</tr>
<tr>
<td>PhaseImbalance</td>
<td>Phase imbalance, Q vs I</td>
<td>0.0</td>
<td>deg</td>
<td>real</td>
<td>(-∞, ∞)</td>
<td></td>
</tr>
<tr>
<td>FCarrier</td>
<td>carrier frequency</td>
<td>474.0MHz</td>
<td>Hz</td>
<td>real</td>
<td>(0, ∞)</td>
<td></td>
</tr>
<tr>
<td>Sensitivity</td>
<td>voltage output sensitivity, ( \text{Vout}/\text{Vin} )</td>
<td>1</td>
<td></td>
<td>real</td>
<td>(-∞, ∞)</td>
<td></td>
</tr>
<tr>
<td>Phase</td>
<td>reference phase in degrees</td>
<td>0.0</td>
<td>deg</td>
<td>real</td>
<td>(-∞, ∞)</td>
<td></td>
</tr>
<tr>
<td>Mode</td>
<td>transmission mode: DVB 2k mode, DVB 8k mode</td>
<td>DVB 2k mode</td>
<td>enum</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OversamplingOption</td>
<td>oversampling ratio option: Ratio 1, Ratio 2, Ratio 4, Ratio 8, Ratio 16, Ratio 32</td>
<td>Ratio 1</td>
<td>S</td>
<td>enum</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GuardInterval</td>
<td>guard interval (frac FFT size)</td>
<td>0.125</td>
<td></td>
<td>real</td>
<td>[0, 1]</td>
<td></td>
</tr>
<tr>
<td>CodeRateHP</td>
<td>high priority stream current code rate: HP 1/2, HP 2/3, HP 3/4, HP 5/6, HP 7/8</td>
<td>HP 1/2</td>
<td></td>
<td>enum</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CodeRateLP</td>
<td>low priority stream current code rate: LP 1/2, LP 2/3, LP 3/4, LP 5/6, LP 7/8</td>
<td>LP 1/2</td>
<td></td>
<td>enum</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MappingMode</td>
<td>signal constellation and mapping: QAM-16, QAM-64</td>
<td>QAM-16</td>
<td></td>
<td>enum</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
DVB-T Components

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Default</th>
<th>Sym</th>
<th>Unit</th>
<th>Type</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alpha</td>
<td>non-uniform factor for DVB-T</td>
<td>1</td>
<td></td>
<td>int</td>
<td></td>
<td>[1, ∞)</td>
</tr>
<tr>
<td>SoftDecision</td>
<td>soft decision viterbi decoding type: Soft, Hard, Cochannel</td>
<td>Soft</td>
<td></td>
<td></td>
<td>enum</td>
<td></td>
</tr>
<tr>
<td>TrunLen</td>
<td>path memory truncation length of Viterbi decoding algorithm, in bytes</td>
<td>10</td>
<td></td>
<td>int</td>
<td></td>
<td>[5, ∞)</td>
</tr>
</tbody>
</table>

Pin Inputs

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Rx_signal</td>
<td>received OFDM signal to be demodulated</td>
<td>timed</td>
</tr>
</tbody>
</table>

Pin Outputs

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>HPBytes_Decoded</td>
<td>decoded bytes after Reed Solomon decoder with high priority</td>
<td>int</td>
</tr>
<tr>
<td>3</td>
<td>LPBytes_Decoded</td>
<td>decoded bytes after Reed Solomon decoder with low priority</td>
<td>int</td>
</tr>
<tr>
<td>4</td>
<td>HPBytes_AfterViterbi</td>
<td>decoded bytes after Viterbi decoder with high priority</td>
<td>int</td>
</tr>
<tr>
<td>5</td>
<td>LPBytes_AfterViterbi</td>
<td>decoded bytes after Viterbi decoder with low priority</td>
<td>int</td>
</tr>
<tr>
<td>6</td>
<td>TPS_Bits</td>
<td>decoded TPS information bits</td>
<td>int</td>
</tr>
<tr>
<td>7</td>
<td>constellation</td>
<td>constellation signal after OFDM demodulation</td>
<td>complex</td>
</tr>
</tbody>
</table>

Notes/Equations

1. This subnetwork model implements the DVB-T RF hierarchical receiver function. The received RF signal is demodulated and fed to the baseband receiver.

The schematic for this subnetwork is shown in Figure 3-13.
The baseband demodulator schematic is shown in Figure 3-14.
Figure 3-13. DTV_DVBTHierReceiver_RF schematic
Figure 3-14. DTV_DVBTHierReceiver Schematic
2. Output Delays

The constellation pin has a delay of 8 OFDM symbols. The first \(8 \times 1705\) (2k mode) or \(8 \times 6817\) (8k mode) constellation output signals are not used for EVM calculation.

DTV_BCHDecoder uses a 68-bit block; so, a Delay component with \(N=60\) is inserted before DTV_DVBTPSDemod.

Pin TPS_Bits has one DVB-T frame (68 OFDM symbols) delay; the first 53 bits at TPS_Bits are not used.

LPBytes_AfterViterbi and HPBytes_AfterViterbi each have 8 OFDM symbol delays.

Inner (or convolutional) decoding is based on an MPEG-2 transport MUX packet. So, the LPBytes_Decoded, HPBytes_Decoded, HPBytes_AfterViterbi, and LPBytes_AfterViterbi delays must be multiples of coded (204) and uncoded (188) transport MUX packet blocks.

The information bytes for one OFDM symbol in HPBytes_AfterViterbi is calculated as follows:

\[
\text{DataPerOFDMSymbol} = 1512 \text{ (2k mode) or } 6048 \text{ (8k mode)} \\
\text{CodedBlockSize} = 204 \\
\text{HPCodeRate} = \frac{1}{2}, \frac{2}{3}, \frac{3}{4}, \frac{5}{6} \text{ or } \frac{7}{8} \\
\text{DelayOFDMSymbols} = 8 \\
\text{HPInfoBytes} = \text{DelayOFDMSymbols} \times \text{DataPerOFDMSymbol} \times \text{HPCodeRate}/8
\]

The block delay of HPBytes_AfterViterbi is \(\text{Delay_{HPAfterViterbi}}\) and calculated as follows:

\[
\text{HPDelayBytes} = \text{DelayOFDMSymbols} \times \text{HPInfoBytes} \\
-(\text{int}(\text{DelayOFDMSymbols} \times \text{HPInfoBytes} \times \text{CodedBlockSize})) \times \text{CodedBlockSize} \\
\text{N}_{\text{HP}} = \text{int}((\text{TrunLen} + \text{LPDelayBytes}) \times \text{CodedBlockSize})/\text{CodedBlockSize} \\
\text{Delay_{HPAfterViterbi}} = 1 + \text{int}((\text{DelayOFDMSymbols} \times \text{HPInfoBytes})/(\text{CodedBlockSize})) + \text{N}_{\text{HP}}.
\]

Block size is 204 bytes.

The information bytes for one OFDM symbol in LPBytes_AfterViterbi is calculated as follows:
DVB-T Components

- DataPerOFDMSymbol = 1512 (2k mode) or 6048 (8k mode)
- CodedBlockSize = 204
- BitsPerQAMmappingLP = 2 for 16-QAM or 4 for 64-QAM
- LPCodeRate = 1/2, 2/3, 3/4, 5/6 or 7/8
- LPInfoBytes = BitsPerQAMmappingLP × DataPerOFDMSymbol × LPCodeRate/8

The block delay of LPBytes_AfterViterbi is calculated as follows:

LPDelayBytes = DelayOFDMSymbols × LPInfoBytes - \( \left( \text{int}(\text{DelayOFDMSymbols} \times \text{LPInfoBytes}/\text{CodedBlockSize}) \right) \times \text{CodedBlockSize} \),

N_LP = \( \text{int}((\text{TrunLen} + \text{LPDelayBytes})/\text{CodedBlockSize}) \),

Delay_LPAfterViterbi = 1 + \( \text{int}((\text{DelayOFDMSymbols} \times \text{LPInfoBytes})/\text{CodedBlockSize}) + N_LP \).

Block size is 204 bytes.

HPBytes_Decoded and LPBytes_Decoded each have 8 OFDM symbols and 11 transport MUX packet delays.

The block delay of HPBytes_Decoded is:

Delay_HPDecoded = 11 + 1 + \( \text{int}((\text{DelayOFDMSymbols} \times \text{HPIInfoBytes})/\text{CodedBlockSize}) + N_HP \).

Block size is 188 bytes.

The block delay of LPBytes_Decoded is:

Delay_LPDecoded = 11 + 1 + \( \text{int}((\text{DelayOFDMSymbols} \times \text{LPIInfoBytes})/\text{CodedBlockSize}) + N_LP \).

Block size is 188 bytes.
3. Receiver functions are implemented according to the DVB-T Standard.

Start of OFDM symbol is detected by the DTV_DVBTOFDMDemod subnetwork (Figure 3-15).

**Figure 3-15. DTV_DVBTOFDMDemod Schematic**

DTV_MLEstimator calculates the correlation based on guard interval;
DTV_LoadFFTBuff selects the index with the maximum correlation value as the start of FFT.

DTV_DVBCh2DEstimator estimates the complex channel impulse responses for continual and scattered pilot subcarriers. Channel impulse responses of the data subcarriers are 2D interpolated based on the estimated CIR.

Each subcarrier value is divided with a complex estimated channel response coefficient (phase tracking, phase synchronization, and equalization). This simple one-tap frequency domain channel response compensation is implemented by DTV_OFDMEqualizer.
DVB-T Components

After equalization, DTV_DVBDemuxOFDMSym demultiplexes 1705 or 6817 subcarriers into 1512 (2k mode) or 6048 (8k mode) data subcarriers and 1 demodulated TPS.

The demodulated 1512 or 6048 data subcarriers (16QAM, and 64QAM modulation) are demapped by DTV_Demapper.

The demodulated bits are deinterleaved by DTV_DVBTHierInnerDeinterlv and divided into high- and low-priority demodulated bits. The demodulated bits are inner decoded (DTV_InnerDecoder), outer de-interleaved (DTV_InterlvInt), outer decoded (DTV_RSDecoder) and descrambled (DTV_ScrambleByte). The fully-decoded high- and low-priority stream bytes are output as HPBytes_Decoded and LPBytes_Decoded.

After inner-decoding, the partially-decoded high- and low-priority stream bytes are output as HPBytes_AfterViterbi and LPBytes_AfterViterbi.

The TPS signal is demodulated by DTV_DVBTPSDemod; the demodulated bits are decoded by DTV_BCHDecoder.

Output pins of this DTV_DVBTHierReceiver subnetwork correspond to those of DTV_DVBTHierSignalSrc.

This hierarchical receiver works uses one OFDM symbol. The number of samples of each OFDM symbol (Rx_signal input pin) is:

• \((1+\text{GuardInterval})\times2048\times\text{OversamplingRatio}\) for 2k mode
• \((1+\text{GuardInterval})\times8192\times\text{OversamplingRatio}\) for 8k mode

The number of signals at the constellation output pin is 1705 (2k mode) or 6817 (8k mode) for one OFDM symbol.
4. Parameter Details

GainImbalance and PhaseImbalance add certain impairments to the ideal output RF signal. Impairments are added in the order described here.

The unimpaired RF I and Q envelope voltages have gain and phase imbalance applied. The RF is given by:

\[
V_3(t) = A \left( V_1(t) \cos(\omega_c t) - g V_2(t) \sin\left(\omega_c t + \frac{\phi \pi}{180}\right) \right)
\]

where \(V_3(t)\) is the in-phase RF envelope, \(V_2(t)\) is the quadrature phase RF envelope, \(g\) is the gain imbalance

\[
g = \frac{\text{GainImbalance}}{20}
\]

and, \(\phi\) (in degrees) is the phase imbalance.

Mode specifies the 2k or 8k transmission mode as defined in DVB-T.

OversamplingOption specifies the transmission signal oversampling ratio 1, 2, 4, 8, 16, 32. If OversamplingOption=Ratio 2, the IFFT size is 4096 for 2k mode or 16384 for 8k mode.

GuardInterval specifies the guard interval (also called cyclic extension) length for each symbol time, as a fraction of the FFT time period. Guard interval types 1/4, 1/8, 1/16 and 1/32 are defined in the DVB-T specification. GuardInterval can be set to any value between 0.0 and 1.0 (not just 1/4, 1/8, 1/16 and 1/32). For proper demodulation, the value must match the guard interval length actually used in the input signal.

CodeRateHP specifies the high-priority stream code rate: 1/2, 2/3, 3/4, 5/6 or 7/8.

CodeRateLP specifies the low-priority stream code rate: 1/2, 2/3, 3/4, 5/6 or 7/8.

MappingMode specifies signal constellations and mapping for the DVB-T hierarchical signal source. Mapping modes 16-QAM and 64-QAM are available.

Alpha is used for signal constellation and mapping as defined in DVB-T specifications. If Alpha=1, 16-QAM or 64-QAM uniform mapping is performed; if Alpha=2 or 4, 16-QAM or 64-QAM in non-uniform mapping is performed.

SoftDecision specifies the Viterbi decoding algorithm mode.

• If SoftDecision=Soft, the Viterbi decoding algorithm is a soft decision decoder and uses channel status information.
DVB-T Components

- If SoftDecision=Hard, the Viterbi decoding algorithm is a hard decision decoder and does not use channel status information.
- If SoftDecision=Cochannel, the Viterbi decoding algorithm is for co-channel measurement with PAL analog TV and digital DVB-T.

TrunLen sets the truncation length (in bytes) in the Viterbi decoding algorithm. The path memory is 8×TrunLen bits in Viterbi decoding algorithm.

References


**DTV_DVBTHierSignalSrc**

**Description**  
DVB-T hierarchical signal source  
**Library**  
DTV, DVB-T  
**Required Licenses**

**Parameters**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Default</th>
<th>Sym</th>
<th>Type</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mode</td>
<td>transmission mode: DVB 2k mode, DVB 8k mode</td>
<td>DVB 2k mode</td>
<td></td>
<td>enum</td>
<td></td>
</tr>
<tr>
<td>OversamplingOption</td>
<td>oversampling ratio option: Ratio 1, Ratio 2, Ratio 4, Ratio 8, Ratio 16, Ratio 32</td>
<td>Ratio 1</td>
<td>S</td>
<td>enum</td>
<td></td>
</tr>
<tr>
<td>GuardInterval</td>
<td>guard interval (frac FFT size)</td>
<td>0.125</td>
<td>real</td>
<td>[0, 1]</td>
<td></td>
</tr>
<tr>
<td>CodeRateHP</td>
<td>high priority stream current code rate: HP 1/2, HP 2/3, HP 3/4, HP 5/6, HP 7/8</td>
<td>HP 1/2</td>
<td></td>
<td>enum</td>
<td></td>
</tr>
<tr>
<td>CodeRateLP</td>
<td>low priority stream current code rate: LP 1/2, LP 2/3, LP 3/4, LP 5/6, LP 7/8</td>
<td>LP 1/2</td>
<td></td>
<td>enum</td>
<td></td>
</tr>
<tr>
<td>MappingMode</td>
<td>signal constellation and mapping: QAM-16, QAM-64</td>
<td>QAM-16</td>
<td></td>
<td>enum</td>
<td></td>
</tr>
<tr>
<td>Alpha</td>
<td>non-uniform factor for DVB-T</td>
<td>1</td>
<td>int</td>
<td>[1, infinity]</td>
<td></td>
</tr>
<tr>
<td>DataTypeHP</td>
<td>Payload data type with high priority: PN9 HP, PN15 HP, FIX4 HP, _4_1_4_0 HP, _8_1_8_0 HP, _16_1_16_0 HP, _32_1_32_0 HP, _64_1_64_0 HP</td>
<td>PN9 HP</td>
<td></td>
<td>enum</td>
<td></td>
</tr>
<tr>
<td>DataTypeLP</td>
<td>Payload data type with low priority: PN9 LP, PN15 LP, FIX4 LP, _4_1_4_0 LP, _8_1_8_0 LP, _16_1_16_0 LP, _32_1_32_0 LP, _64_1_64_0 LP</td>
<td>PN15 LP</td>
<td></td>
<td>enum</td>
<td></td>
</tr>
</tbody>
</table>
DVB-T Components

Pin Outputs

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>DVBT_signal</td>
<td>DVB-T signal</td>
<td>complex</td>
</tr>
<tr>
<td>2</td>
<td>HPBytes_Uncoded</td>
<td>information bytes of transport packet with high priority</td>
<td>int</td>
</tr>
<tr>
<td>3</td>
<td>LPBytes_Uncoded</td>
<td>information bytes of transport packet with low priority</td>
<td>int</td>
</tr>
<tr>
<td>4</td>
<td>HPBytes_BeforeCC</td>
<td>information bytes before convolutional coder with high priority</td>
<td>int</td>
</tr>
<tr>
<td>5</td>
<td>LPBytes_BeforeCC</td>
<td>information bytes before convolutional coder with low priority</td>
<td>int</td>
</tr>
<tr>
<td>6</td>
<td>TPS_Bits</td>
<td>TPS information bits</td>
<td>int</td>
</tr>
<tr>
<td>7</td>
<td>constellation</td>
<td>constellation signal before IFFT</td>
<td>complex</td>
</tr>
</tbody>
</table>

Notes/Equations
1. This subnetwork model is used to generate the DVB-T hierarchical signal source. Each firing, one OFDM symbol is generated.

The schematic for this subnetwork is shown in Figure 3-16.
Note To use the 8 transport MUX packet structure according to the DVB-T standard, HP_stream and LP_stream with 8 transport MUX packets can be made active (HP_stream1, LP_stream1, B2 and B3 must be made inactive). However, the signal source with 8 transport MUX packets will consume much more memory during simulation.

2. The format of the hierarchical signal source follows the DVB-T specification. Figure 3-17 is functional block diagram of the DVB-T baseband system.

The DTV_DVBTHierSignalSrc includes: scrambler (DTV_ScrambleByte); outer coder (DTV_RSCoder); outer interleaver (DTV_InterlvInt); inner coder (DTV_PuncConvCoder); inner interleaver (DTV_DVBTHierInnerInterlv); mapper (DTV_Mapper); and, frame adaptation, OFDM, and guard interval insertion (in DTV_DVBT_OFDMMod subnetwork).

Figure 3-17. Baseband System Functional Block Diagram

This hierarchical signal source works per one OFDM symbol: 1705 (2k mode) or 6817 (8k mode) constellations are output. The number of samples for each OFDM symbol (DVBT_signal output pin) is:

- (1+GuardInterval)×2048×OversamplingRatio for 2k mode
- (1+GuardInterval)×8192×OversamplingRatio for 8k mode
DVB-T Components

3. Parameter Details

Mode specifies the 2k or 8k transmission mode as defined in the DVB-T specification.

OversamplingOption specifies the transmission signal oversampling ratio 1, 2, 4, 8, 16, 32. If OversamplingOption=2, the IFFT size is 4096 (2k mode) or 16384 (8k mode).

GuardInterval specifies the guard interval (also called cyclic extension) length for each symbol time, as a fraction of the FFT time period. Guard interval types 1/4, 1/8, 1/16 and 1/32 are defined in the DVB-T specification. GuardInterval can be set to any value between 0.0 and 1.0 (not just 1/4, 1/8, 1/16 and 1/32). For proper demodulation, the value must match the guard interval length actually used in the input signal.

CodeRateHP specifies the high priority stream code rate: 1/2, 2/3, 3/4, 5/6 or 7/8.

CodeRateLP specifies the low priority stream code rate: 1/2, 2/3, 3/4, 5/6 or 7/8.

MappingMode specifies the DVB-T hierarchical signal source constellations: 16-QAM or 64-QAM.

Alpha is used for signal constellation and mapping (based on MappingMode).

• If Alpha=1, uniform mapping of 16-QAM or 64-QAM is used.
• If Alpha=2 or 4, non-uniform mapping of 16-QAM or 64-QAM is used as defined in DVB-T specification.

For the DataTypeHP parameter:

• if PN9 HP is selected, a 511-bit pseudo-random test pattern is generated according to CCITT Recommendation O.153
• if PN15 HP is selected, a 32767-bit pseudo-random test pattern is generated according to CCITT Recommendation O.151
• if FIX4 HP is selected, a zero-stream is generated
• if x_1_x_0 HP is selected (x equals 4, 8, 16, 32, or 64) a periodic bit stream is generated, with the period being 2x. In one period, the first x bits are 1s and the second x bits are 0s.
For the DataTypeLP parameter:

- if PN9 LP is selected, a 511-bit pseudo-random test pattern is generated according to CCITT Recommendation O.153
- if PN15 LP is selected, a 32767-bit pseudo-random test pattern is generated according to CCITT Recommendation O.151
- if FIX4 LP is selected, a zero-stream is generated
- if x_1_x_0 is selected (x equals 4, 8, 16, 32, or 64) a periodic bit stream is generated, with the period being 2x. In one period, the first x bits are 1s and the second x bits are 0s.

References

DVB-T Components

**DTV\_DVBTHierSignalSrc\_RF**

**Description**  
DVB-T RF hierarchical signal source

**Library**  
DTV, DVB-T

### Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Default</th>
<th>Sym</th>
<th>Unit</th>
<th>Type</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>ROut</td>
<td>output resistance</td>
<td>DefaultROut</td>
<td>Ohm</td>
<td>real</td>
<td>(0, ∞)</td>
<td></td>
</tr>
<tr>
<td>RTemp</td>
<td>physical temperature, in degrees C</td>
<td>DefaultRTemp</td>
<td>Celsius</td>
<td>real</td>
<td>[-273.15, ∞)</td>
<td></td>
</tr>
<tr>
<td>Noise</td>
<td>Enable thermal noise? NO, YES</td>
<td>NO</td>
<td>enum</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FCarrier</td>
<td>carrier frequency</td>
<td>474.0MHz</td>
<td>Hz</td>
<td>real</td>
<td>(0, ∞)</td>
<td></td>
</tr>
<tr>
<td>Power</td>
<td>modulator output power</td>
<td>40mW</td>
<td>W</td>
<td>real</td>
<td>(0, ∞)</td>
<td></td>
</tr>
<tr>
<td>Bandwidth</td>
<td>bandwidth (2048/224.0)MHz</td>
<td>Hz</td>
<td>Hz</td>
<td>real</td>
<td>(0, ∞)</td>
<td></td>
</tr>
<tr>
<td>MirrorSpectrum</td>
<td>Mirror spectrum about carrier? NO, YES</td>
<td>NO</td>
<td>enum</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GainImbalance</td>
<td>Gain imbalance, Q vs I</td>
<td>0.0</td>
<td>dB</td>
<td>real</td>
<td>(-∞, ∞)</td>
<td></td>
</tr>
<tr>
<td>PhaseImbalance</td>
<td>Phase imbalance, Q vs I</td>
<td>0.0</td>
<td>deg</td>
<td>real</td>
<td>(-∞, ∞)</td>
<td></td>
</tr>
<tr>
<td>I_OriginOffset</td>
<td>I origin offset (percent)</td>
<td>0.0</td>
<td>real</td>
<td>(-∞, ∞)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Q_OriginOffset</td>
<td>Q origin offset (percent)</td>
<td>0.0</td>
<td>real</td>
<td>(-∞, ∞)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IQ_Rotation</td>
<td>IQ rotation</td>
<td>0.0</td>
<td>deg</td>
<td>real</td>
<td>(-∞, ∞)</td>
<td></td>
</tr>
<tr>
<td>Mode</td>
<td>transmission mode: DVB 2k mode, DVB 8k mode</td>
<td>DVB 2k mode</td>
<td>enum</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OversamplingOption</td>
<td>oversampling ratio option: Ratio 1, Ratio 2, Ratio 4, Ratio 8, Ratio 16, Ratio 32</td>
<td>Ratio 1</td>
<td>S</td>
<td>enum</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GuardInterval</td>
<td>guard interval (frac FFT size)</td>
<td>0.125</td>
<td>real</td>
<td>[0, 1]</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
This subnetwork model integrates an RF modulator and a baseband hierarchical signal source based on the DVB-T specification. The baseband hierarchical signal source is designed to support different code rates and mapping modes, as well as various data types for high and low priority streams.

### Pin Outputs

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>RF_signal</td>
<td>RF signal</td>
<td>timed</td>
</tr>
<tr>
<td>2</td>
<td>DVBT_signal</td>
<td>DVB-T signal</td>
<td>complex</td>
</tr>
<tr>
<td>3</td>
<td>HPBytes_Uncoded</td>
<td>information bytes of transport packet with high priority</td>
<td>int</td>
</tr>
<tr>
<td>4</td>
<td>LPBytes_Uncoded</td>
<td>information bytes of transport packet with low priority</td>
<td>int</td>
</tr>
<tr>
<td>5</td>
<td>HPBytes_BeforeCC</td>
<td>information bytes before convolutional coder with high priority</td>
<td>int</td>
</tr>
<tr>
<td>6</td>
<td>LPBytes_BeforeCC</td>
<td>information bytes before convolutional coder with low priority</td>
<td>int</td>
</tr>
<tr>
<td>7</td>
<td>TPS_Bits</td>
<td>TPS information bits</td>
<td>int</td>
</tr>
<tr>
<td>8</td>
<td>constellation</td>
<td>constellation signal before IFFT</td>
<td>complex</td>
</tr>
</tbody>
</table>

### Notes/Equations

1. This subnetwork model integrates an RF modulator and a baseband hierarchical signal source based on the DVB-T specification. The baseband
DVB-T Components

Hierarchical signal is fed to the RF modulator; after RF modulation the timed signal is output.

The schematic for this subnetwork is shown in Figure 3-18.

Figure 3-18. DTV_DVBTHierSignalSrc_RF Schematic

2. The format of DVB-T signal source follows the DVB-T specification. Figure 3-19 is a functional block diagram of a DVB-T baseband system.

Figure 3-19. Baseband System Block Diagram

3. The schematic for the hierarchical signal source is shown in Figure 3-20. The DTV_DVBTHierSignalSrc includes: scrambler (DTV_ScrambleByte); outer coder (DTV_RSCoder); outer interleaver (DTV_InterlvInt); inner coder (DTV_PuncConvCoder); inner interleaver (DTV_DVBTHierInnerInterlv); mapper (DTV_Mapper); and, frame adaptation, OFDM, and guard interval insertion (in DTV_DVBTOFDMMod subnetwork).
Note To use the 8 transport MUX packet structure, HP_stream and LP_stream with 8 transport MUX packets must be active, and HP_stream1, LP_stream1, B2, and B3 must be inactive. A signal source with 8 transport MUX packets will consume more memory during simulation.

Figure 3-20. DTV_DVBTHierSignalSrc Schematic

This hierarchical signal source works per each OFDM symbol: 1705 (2k mode) or 6817 (8k mode) constellations are output. DVBT_signal pin outputs samples for each OFDM symbol:

- \((1 + \text{GuardInterval}) \times 2048 \times \text{OversamplingRatio}\) for 2k mode
- \((1 + \text{GuardInterval}) \times 8192 \times \text{OversamplingRatio}\) for 8k mode
4. Parameter Details

FCarrier defines the RF frequency for the DVB signal.

Power defines the power level for FCarrier.

Bandwidth indicates signal bandwidth.

MirrorSpectrum (when set to YES) conjugates the input signal before any other processing is done.

The GainImbalance, PhaseImbalance, I_OriginOffset, Q_OriginOffset, and IQ_Rotation parameters add certain impairments to the ideal output RF signal. Impairments are added in the order described here.

The unimpaired RF I and Q envelope voltages have gain and phase imbalance applied. The RF is given by:

\[ V_3(t) = A \left( V_1(t) \cos(\omega_c t) - gV_2(t) \sin\left(\omega_c t + \frac{\phi \pi}{180}\right) \right) \]

where \( A \) is a scaling factor that depends on the Power and ROut parameters specified by the user, \( V_1(t) \) is the in-phase RF envelope, \( V_2(t) \) is the quadrature phase RF envelope, \( g \) is the gain imbalance, and \( \phi \) (in degrees) is the phase imbalance.

The signal \( V_3(t) \) is rotated by IQ_Rotation degrees; I_OriginOffset and Q_OriginOffset are then applied to the rotated signal. Note that the amounts specified are percentages with respect to the output rms voltage. The output rms voltage is given by \( \sqrt{2 \times ROut \times Power} \).

Mode specifies the 2k or 8k transmission mode as defined in the DVB-T specification.

OversamplingOption specifies the transmission signal oversampling ratio 1, 2, 4, 8, 16, 32. If OversamplingOption=2, the IFFT size is 4096 (2k mode) or 16384 (8k mode).

GuardInterval specifies the guard interval (also called cyclic extension) length for each symbol time, as a fraction of the FFT time period. Guard interval types 1/4, 1/8, 1/16 and 1/32 are defined in the DVB-T specification. GuardInterval can be set to any value between 0.0 and 1.0 (not just 1/4, 1/8, 1/16 and 1/32). For
proper demodulation, the value must match the guard interval length actually used in the input signal.

CodeRateHP specifies the high priority stream code rate: 1/2, 2/3, 3/4, 5/6 or 7/8.

CodeRateLP specifies the low priority stream code rate: 1/2, 2/3, 3/4, 5/6 or 7/8.

MappingMode specifies signal constellation and mapping modes: 16-QAM or 64-QAM.

Alpha is used for signal constellation and mapping as defined in DVB-T specifications. If Alpha=1, 16-QAM or 64-QAM uniform mapping is performed; if Alpha=2 or 4, 16-QAM or 64-QAM in non-uniform mapping is performed.

For the DataTypeHP parameter:

if PN9 is selected, a 511-bit pseudo-random test pattern is generated according to CCITT Recommendation 0.153.

if PN15 is selected, a 32767-bit pseudo-random test pattern is generated according to CCITT Recommendation 0.151.

if FIX4 is selected, a zero-stream is generated.

if x_1_x_0 is selected, where x equals 4, 8, 16, 32, or 64, a periodic bit stream is generated, with the period being 2x. In one period, the first x bits are 1s and the second x bits are 0s.

For the DataTypeLP parameter:

if PN9 is selected, a 511-bit pseudo-random test pattern is generated according to CCITT Recommendation 0.153.

if PN15 is selected, a 32767-bit pseudo-random test pattern is generated according to CCITT Recommendation 0.151.

if FIX4 is selected, a zero-stream is generated.

if x_1_x_0 is selected, where x equals 4, 8, 16, 32, or 64, a periodic bit stream is generated, with the period being 2x. In one period, the first x bits are 1s and the second x bits are 0s.

References

**DVB-T Components**

**DTV_DVBTImpulseNoise**

**Description**  Impulse interference Noise  
**Library**  DTV, DVB-T

**Parameters**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Default</th>
<th>Unit</th>
<th>Type</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test_No</td>
<td>test mode: Test_No_1, Test_No_2, Test_No_3, Test_No_4, Test_No_5, Test_No_6</td>
<td>Test_No_3</td>
<td>enum</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NoisePower</td>
<td>power of impulse interference noise</td>
<td>1.0W</td>
<td>W</td>
<td>real</td>
<td>(-∞, ∞)</td>
</tr>
<tr>
<td>TStep</td>
<td>time step</td>
<td>0.19375usec</td>
<td>sec</td>
<td>real</td>
<td>(0, ∞)</td>
</tr>
<tr>
<td>FCarrier</td>
<td>carrier frequency</td>
<td>474.0MHz</td>
<td>Hz</td>
<td>real</td>
<td>(0, ∞)</td>
</tr>
<tr>
<td>RefR</td>
<td>resistance</td>
<td>500Ohm</td>
<td>Ohm</td>
<td>real</td>
<td>(0, ∞)</td>
</tr>
</tbody>
</table>

**Pin Outputs**

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Impulse_Interference</td>
<td>impulse interference noise</td>
<td>timed</td>
</tr>
</tbody>
</table>

**Notes/Equations**

1. This model is used to generate impulse interference that is caused by automotive ignition systems, domestic appliance switches, and electric motors. The impulse interference consists of a number of bursts that are generated by gating a Gaussian noise source of power P.

The schematic for this impulse noise generator is shown in Figure 3-21.
2. The impulse noise test signal is illustrated in Figure 3-22.

From Figure 3-22, the separation of bursts is fixed at 10 msec, and the pulse duration is fixed at 250 nsec.
DVB-T Components

**Note**  This model supports fixed spacing between pulses instead of variable spacing described in 11.13.3 of Ref. [1]. Fixed spacing is dependent on test types detailed in note 3.

3. This model supports 6 tests in which the number of pulses per burst and pulse spacing vary according to Test No as listed in Table 3-12.

Table 3-12. Impulse Noise Tests and Pulse Parameters

<table>
<thead>
<tr>
<th>Test_No</th>
<th>Pulses per Burst</th>
<th>Pulse Spacing (usec)</th>
<th>Burst Duration (usec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0.25</td>
<td>0.25</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>20</td>
<td>40</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>20</td>
<td>80</td>
</tr>
<tr>
<td>4</td>
<td>12</td>
<td>12</td>
<td>144</td>
</tr>
<tr>
<td>5</td>
<td>20</td>
<td>1.5</td>
<td>30</td>
</tr>
<tr>
<td>6</td>
<td>40</td>
<td>0.75</td>
<td>30</td>
</tr>
</tbody>
</table>

Noise power is the Gaussian noise power that is gated by pulses. It can be measured by closing pulsing generators labeled R1, R2, R3 in Figure 3-21.

**References**

**DTV_DVBTInnerDeInterlv**

**Description**  Inner de-interleaver for QPSK, 16QAM and 64QAM non-hierarchical transmission modes

**Library**  DTV, DVB-T

**Required Licenses**

**Parameters**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Default</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mode</td>
<td>transmission mode: DVB 2k mode, DVB 8k mode</td>
<td>DVB 2k mode</td>
<td>enum</td>
</tr>
<tr>
<td>MappingMode</td>
<td>signal constellations and mapping: QPSK, QAM-16, QAM-64</td>
<td>QAM-16</td>
<td>enum</td>
</tr>
</tbody>
</table>

**Pin Inputs**

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Symbols</td>
<td>input of inner de-interleaver</td>
<td>anytype</td>
</tr>
</tbody>
</table>

**Pin Outputs**

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>Bits</td>
<td>output bit stream of inner de-interleaver</td>
<td>anytype</td>
</tr>
</tbody>
</table>

**Notes/Equations**

1. This subnetwork model is used to perform inner de-interleaving QPSK, 16QAM and 64QAM non-hierarchical transmission modes. Functions are performed by an if-else structure.

   The schematic for this subnetwork is shown in Figure 3-23.
DVB-T Components

2. The Mode parameter is used to specify the 2k mode or the 8k mode.

3. The MappingMode parameter is used to specify QPSK, QAM-16 or QAM-64.

References

Description  Inner interleaver for QPSK, 16QAM, 64QAM non-hierarchical transmission modes  
Library  DTV, DVB-T  
Required Licenses

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Default</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mode</td>
<td>transmission mode: DVB 2k mode, DVB 8k mode</td>
<td>DVB 2k mode</td>
<td>enum</td>
</tr>
<tr>
<td>MappingMode</td>
<td>signal constellations and mapping: QPSK, QAM-16, QAM-64</td>
<td>QAM-16</td>
<td>enum</td>
</tr>
</tbody>
</table>

Pin Inputs

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Bits</td>
<td>Bit stream prior to inner interleaver</td>
<td>anytype</td>
</tr>
</tbody>
</table>

Pin Outputs

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>Symbols</td>
<td>output of inner interleaver</td>
<td>anytype</td>
</tr>
</tbody>
</table>

Notes/Equations

1. This subnetwork model is used to perform inner interleaving QPSK, 16QAM and 64QAM. Functions can be performed by an if-else structure. The schematic for this subnetwork is shown in Figure 3-24.
DVB-T Components

Figure 3-24. DTV_DVBTInnerInterlv Schematic

2. The Mode parameter is used to specify the 2k mode or the 8k mode.
3. The MappingMode parameter is used to specify QPSK, QAM-16 or QAM-64.

References

DTV_DVBTOFDMDemod

Description  DVB-T OFDM de-modulator
Library      DTV, DVB-T
Required Licenses

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Default</th>
<th>Sym</th>
<th>Type</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mode</td>
<td>transmission mode: DVB 2k mode, DVB 8k mode</td>
<td>DVB 2k mode</td>
<td>enum</td>
<td></td>
<td></td>
</tr>
<tr>
<td>OversamplingOption</td>
<td>oversampling ratio option: Ratio 1, Ratio 2, Ratio 4, Ratio 8, Ratio 16, Ratio 32</td>
<td>Ratio 1</td>
<td>S</td>
<td>enum</td>
<td></td>
</tr>
<tr>
<td>GuardInterval</td>
<td>guard interval (frac FFT size)</td>
<td>0.125</td>
<td>real</td>
<td></td>
<td>[0, 1]</td>
</tr>
</tbody>
</table>

Pin Inputs

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>input</td>
<td>received OFDM symbol</td>
<td>complex</td>
</tr>
</tbody>
</table>

Pin Outputs

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>data</td>
<td>information on data subcarriers</td>
<td>complex</td>
</tr>
<tr>
<td>3</td>
<td>TPS</td>
<td>information on TPS subcarriers</td>
<td>complex</td>
</tr>
<tr>
<td>4</td>
<td>CIR</td>
<td>estimated channel impulse response on data subcarriers</td>
<td>complex</td>
</tr>
<tr>
<td>5</td>
<td>constellation</td>
<td>demodulated constellation on all active subcarriers</td>
<td>complex</td>
</tr>
</tbody>
</table>

Notes/Equations

1. This subnetwork model is used to implement OFDM demodulation for DVB-T. The schematic for this subnetwork is shown in Figure 3-25.
2. The data, TPS, and constellation output pins correspond to DTV_DVBTOFDMMod pins; the CIR pin outputs the estimated channel impulse response of each data subcarrier, which is used as channel status information in the soft Viterbi decoding algorithm.

Each firing, the input pin consumes 
\[(1+\text{GuardInterval}) \times 2048 \times \text{OversamplingRatio}\] of 2k mode or 
\[(1+\text{GuardInterval}) \times 8192 \times \text{OversamplingRatio}\] of 8k mode tokens. The data and CIR pins output 1512 (2k mode) or 6048 (8k mode) complex signals; the constellation pin outputs 1705 (2k mode) or 6817 (8k mode) demodulated complex signals; the TPS pin outputs one received TPS signal.

DTV_DVBTOFDMDemod has a delay of 8 OFDM symbols when the received OFDM signal is demodulated; so, the output tokens for the first 8 OFDM symbols in each output pin should be discarded.

3. De-multiplex demodulated subcarriers into TPS, data, and pilots (continual and scattered).
Figure 3-26 illustrates the frame structure. IntState of Length determines the number of CP (continual pilot) and TPS in each symbol; CP and TPS positions are determined according to Table 3-13 and Table 3-14, respectively.

PRBS sequences in each segment are generated according to Figure 3-27; the initial PRBS register setting is 1111111111. The PRBS is initialized so that the first output bit from the PRBS coincides with the first active carrier. A new value is generated by the PRBS on each carrier in each segment (whether or not it is a pilot).

Positions of the corresponding scattered pilots are generated as follows. For the symbol of index $l$ (ranging from 0 to 67), carriers for which index $k$ belongs to subset

$$\{k = K_{\text{min}} + 3 \times l \mod 4 + 12p \mid p \text{ integer}, p \geq 0, k \in [K_{\text{min}}, K_{\text{max}}]\}$$

are scattered pilots positions.

where $p$ is an integer that takes all possible values 0, provided the resulting value for $k$ does not exceed the valid range $[K_{\text{min}}, K_{\text{max}}]$; $K_{\text{min}}=0$, and $K_{\text{max}}=1705$ (2k mode) or 6817 (8k mode). SPperiod, SPstart, SPoffset, and SPphase parameters control the scattered pilots positions.

After determining the CP, TPS, and SP positions in each symbol and the value of the PRBS sequence in all active carriers in the symbol, this subnetwork demultiplexes input data into TPS and TSP data. According to the TPS position and the PRBS sequence, one transmitted TPS bit is determined by the mean value of the TPS positions.

$$TPS = \frac{1}{N_{\text{TPS}}} \sum_{l=1}^{N_{\text{TPS}}} \frac{x[\text{TPS position}[l]]}{\text{PilotValue}[\text{TPS position}[l]]}$$

where $N_{\text{TPS}}$ is the number of TPS in one symbol; PilotValue is the PRBS sequence in symbol; $x[i]$ is the input data.

The other positions (other than TPS, CP, and SP) in one segment are the TSP data positions.

$$\text{Data}[i] = x[i]$$
### Table 3-13. Continual Pilot Carrier Indices

<table>
<thead>
<tr>
<th>Continual pilot carrier positions (index number k)</th>
<th>8k mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>2k mode</td>
<td>8k mode</td>
</tr>
<tr>
<td>0 48 54 87 141 156 192 201 255 279 282 333 432 450 483 525 531</td>
<td>0 48 54 87 141 156 192 201 255 279 282 333 432 450 483 525 531</td>
</tr>
<tr>
<td>618 636 714 759 765 780 804 873 888 918 939 942 969 984 1050</td>
<td>618 636 714 759 765 780 804 873 888 918 939 942 969 984 1050</td>
</tr>
<tr>
<td>1101 1107 1110 1137 1140 1146 1206 1269 1323 1377 1491 1683 1704</td>
<td>1101 1107 1110 1137 1140 1146 1206 1269 1323 1377 1491 1683 1704</td>
</tr>
</tbody>
</table>

TPS pilots and continual pilots between K\(_{\text{ref}}\) and K\(_{\text{ref}}\) are not indicated.

- ● Boosted pilot
- ○ Data

### Figure 3-26. Frame Structure

### Table 3-14. TPS Carrier Indices

<table>
<thead>
<tr>
<th>Continual Pilot Carrier Indices</th>
<th>8k mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>2k mode</td>
<td>8k mode</td>
</tr>
<tr>
<td>34 50 209 346 413 569 595 688 790 901 1073 1219 1262 1286</td>
<td>34 50 209 346 413 569 595 688 790 901 1073 1219 1262 1286</td>
</tr>
<tr>
<td>1469 1594 1687</td>
<td>1469 1594 1687</td>
</tr>
</tbody>
</table>

### Table 3-14. TPS Carrier Indices

<table>
<thead>
<tr>
<th>Continual Pilot Carrier Indices</th>
<th>8k mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>2k mode</td>
<td>8k mode</td>
</tr>
<tr>
<td>34 50 209 346 413 569 595 688 790 901 1073 1219 1262 1286</td>
<td>34 50 209 346 413 569 595 688 790 901 1073 1219 1262 1286</td>
</tr>
<tr>
<td>1469 1594 1687</td>
<td>1469 1594 1687</td>
</tr>
</tbody>
</table>
4. Parameter Details

Mode is used to select transmission 2k or 8k mode as defined in DVB-T.

OversamplingOption indicates the oversampling ratio of transmission signal. Oversampling ratios 1, 2, 4, 8, 16, 32 are supported. If OversamplingOption=Ratio 2, the IFFT size is 4096 for the 2k mode or 16384 for the 8k mode.

GuardInterval specifies the guard interval (also called cyclic extension) length for each symbol time, as a fraction of the FFT time period. The value must match the guard interval length actually used in the input signal in order for the proper demodulation.

References

DVB-T Components

**DTV_DVBTOFDMMod**

**Description**  DVB-T OFDM modulator

**Library**  DTV, DVB-T

**Required Licenses**

**Parameters**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Default</th>
<th>Sym</th>
<th>Type</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mode</td>
<td>transmission mode: DVB 2k mode, DVB 8k mode</td>
<td>DVB 2k mode</td>
<td>enum</td>
<td></td>
<td></td>
</tr>
<tr>
<td>OversamplingOption</td>
<td>oversampling ratio option: Ratio 1, Ratio 2, Ratio 4, Ratio 8, Ratio 16, Ratio 32</td>
<td>Ratio 1</td>
<td>S</td>
<td>enum</td>
<td></td>
</tr>
<tr>
<td>GuardInterval</td>
<td>guard interval (frac FFT size)</td>
<td>0.125</td>
<td>real</td>
<td></td>
<td>[0, 1]</td>
</tr>
</tbody>
</table>

**Pin Inputs**

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>data</td>
<td>data subcarriers</td>
<td>complex</td>
</tr>
<tr>
<td>2</td>
<td>TPS</td>
<td>TPS signals</td>
<td>complex</td>
</tr>
</tbody>
</table>

**Pin Outputs**

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>signal</td>
<td>OFDM symbol</td>
<td>complex</td>
</tr>
<tr>
<td>4</td>
<td>constellation</td>
<td>mapping signal before OFDM modulation</td>
<td>complex</td>
</tr>
</tbody>
</table>

**Notes/Equations**

1. This subnetwork model is used to implement OFDM modulation for DVB-T. The schematic for this subnetwork is shown in Figure 3-28.
DTV_DVBTOFDMMod 3-89

Figure 3-28. DTV_DVBTOFDMMod Schematic

DTV_DVBMuxOFDMSym multiplexes TPS, data, and pilots (continual and scattered) into one OFDM symbol in the frequency domain and outputs (at the constellation pin) 1705 (2k mode) or 6817 (8k mode) subcarriers each firing.

DTV_DVBLoadIFFTBuff loads transmission data (1705 or 6817) into the IFFT buffer and inserts zeros into the other subcarriers. After IFFT, the frequency domain signals are converted into the time domain.

DTV_InsertGuard inserts a cyclic prefix as the guard interval before the IFFT signal. One OFDM symbol is then generated.

Each firing: TPS pin consumes one input token; the data pin fires 1512 (2k mode) or 6048 (8k mode) input tokens; constellation pin outputs (1705 or 6817) mapping signals and signal pin output

\[(1+\text{GuardInterval}) \times 2048 \times \text{OversamplingRatio} \times \text{bitrate} \quad \text{(2k) or} \quad (1+\text{GuardInterval}) \times 8192 \times \text{OversamplingRatio} \times \text{bitrate} \] complex signals.

2. Multiplex TPS, data and pilots (continual pilots and scattered pilots)

The OFDM frame structure is illustrated in Figure 3-29. The number of IntState of Length determines the number of CP (continual pilot) and TPS in each symbol; the CP and TPS positions are determined according to Table 3-15 and Table 3-16, respectively.

PRBS sequences in each segment are generated according to Figure 3-30; the initial sets of PRBS register is 11111111111. The PRBS in initialized so that the first output bit from the PRBS coincides with the first active carrier. A new value is generated by the PRBS on each carrier in each segment (whether or not it is a pilot).
Positions of the corresponding scattered pilots are generated as follows. For the symbol of index \( l \) (ranging from 0 to 67), carriers for which index \( k \) belongs to subset

\[
\{ k = K_{\text{min}} + 3 \times l \mod 4 + 12p \mid p \text{ integer, } p \geq 0, k \in [K_{\text{min}}, K_{\text{max}}] \}
\]

are scattered pilot positions.

where \( p \) is an integer that takes all possible values \( 0 \), provided the resulting value for \( k \) does not exceed the valid range \([K_{\text{min}}, K_{\text{max}}]\); \( K_{\text{min}} = 0 \), and \( K_{\text{max}} = 1705 \) (2k mode) or 6817 (8k mode). SP period, SP start, and SP offset parameters control the scattered pilots positions.

After determining the CP, TPS, and SP positions in each symbol and the value of the PRBS sequence in all active carriers in one symbol, the model demultiplexes the input data into TPS and TSP data. According to the TPS position and the PRBS sequence, one TPS bit is output in each TPS position.

\[
x[\text{TPS position}[l]] = \text{PilotValue}[\text{TPS position}[l]] \times \text{TPS}
\]

where \( N_{\text{TPS}} \) is the number of TPS in one symbol; \( \text{PilotValue} \) is the PRBS sequence in symbol; \( x[i] \) is the OFDM symbol data.

Except for TPS, CP, and SP positions, the remaining positions in one segment are the TSP data positions.

\[
x[i] = \text{Data}[i]
\]

---

**Figure 3-29. Frame Structure**

TPS pilots and continued pilots between \( K_{\text{min}} \) and \( K_{\text{max}} \) are not indicated.

- • boosted pilot
- ○ data
### Table 3-15. Continual Pilot Carrier Indices

<table>
<thead>
<tr>
<th>Continual pilot carrier positions (index number k)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>2k mode</strong></td>
</tr>
<tr>
<td>0 48 54 87 141 156 192 201 255 279 282 333 432 450 483 525 531 618 636 714 759 765 780 804 873 888 918 939 942 969 984 1050 1101 1107 1110 1137 1140 1146 1206 1269 1323 1377 1491 1683 1704</td>
</tr>
</tbody>
</table>

### Table 3-16. TPS Carrier Indices

<table>
<thead>
<tr>
<th><strong>2k mode</strong></th>
<th><strong>8k mode</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>34 50 209 346 413 569 595 688 790 901 1073 1219 1262 1286 1469 1594 1687</td>
<td>34 50 209 346 413 569 595 688 790 901 1073 1219 1262 1286 1469 1594 1687</td>
</tr>
</tbody>
</table>

### Figure 3-30. PRBS Sequence Generation

#### 3. Parameter Details

Mode specifies the 2k or 8k transmission mode as defined in DVB-T.

OversamplingOption specifies the oversampling ratio of transmission signal: 1, 2, 4, 8, 16, or 32. For example, if OversamplingOption = Ratio 2, the IFFT size is 4096 (2k mode) or 16384 (8k mode).

GuardInterval specifies the guard interval (also called cyclic extension) length for each symbol time, as a fraction of the FFT time period. For proper demodulation, the value must match the guard interval length that is actually used in the input signal.
DVB-T Components

References

# DTV_DVBTPS

**Description**  Transmission parameter signal information bits  
**Library**  DTV, DVB-T  
**Class**  SDFDTV_DVBTPS

## Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Default</th>
<th>Type</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Length</td>
<td>length of TPS information bits</td>
<td>54</td>
<td>int</td>
<td>{54}</td>
</tr>
<tr>
<td>InitiBit</td>
<td>initialization bit for DBPSK modulation (1 bit)</td>
<td>1</td>
<td>int</td>
<td>{0, 1}</td>
</tr>
<tr>
<td>Sync</td>
<td>synchronization word (16 bits): W0, W1</td>
<td>W0</td>
<td>enum</td>
<td></td>
</tr>
<tr>
<td>LengthIndicator</td>
<td>TPS length indicator (6 bits)</td>
<td>0 1 0 1 1</td>
<td>int array</td>
<td>&quot;0 1 0 1 1 1&quot;</td>
</tr>
<tr>
<td>FrameNumber</td>
<td>frame number in one super frame (2 bits): F1, F2, F3, F4</td>
<td>F1</td>
<td>enum</td>
<td></td>
</tr>
<tr>
<td>Constellation</td>
<td>modulation scheme in DVB-T (2 bits): QPSK, QAM 16, QAM 64, reserved M</td>
<td>QPSK</td>
<td>enum</td>
<td></td>
</tr>
<tr>
<td>HierarchyInform</td>
<td>signaling format for alpha values (3 bits): Non hierarchical, alpha 1, alpha 2, alpha 4, reserved H1, reserved H2, reserved H3, reserved H4</td>
<td>Non hierarchical</td>
<td>enum</td>
<td></td>
</tr>
<tr>
<td>CodeRateHP</td>
<td>high priority stream current code rate (3 bits): HP 1/2, HP 2/3, HP 3/4, HP 5/6, HP 7/8, Reserved C1 HP, Reserved C2 HP, Reserved C3 HP</td>
<td>HP 1/2</td>
<td>enum</td>
<td></td>
</tr>
</tbody>
</table>
DVB-T Components

This model is used to generate 54 bits of TPS information; refer to Table 3-17 for bit assignments.

Table 3-17. PS Signalling Information and Format

<table>
<thead>
<tr>
<th>Bit No.</th>
<th>Format</th>
<th>Purpose/Content</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_0$</td>
<td></td>
<td>Initialization bit for differential 2_PSK modulation</td>
</tr>
<tr>
<td>$S_1$ to $S_{16}$</td>
<td>0011010111101110 or 1100101000010001</td>
<td>Synchronization word</td>
</tr>
<tr>
<td>$S_{17}$ to $S_{22}$</td>
<td>010 111</td>
<td>Length indicator</td>
</tr>
<tr>
<td>$S_{23}$ to $S_{24}$</td>
<td>See Table 3-18</td>
<td>Frame number</td>
</tr>
<tr>
<td>$S_{25}$ to $S_{26}$</td>
<td>See Table 3-19</td>
<td>Constellation</td>
</tr>
<tr>
<td>$S_{27}$ to $S_{29}$</td>
<td>See Table 3-20</td>
<td>Hierarchy information</td>
</tr>
<tr>
<td>$S_{30}$ to $S_{32}$</td>
<td>See Table 3-21</td>
<td>Code rate, HP stream</td>
</tr>
<tr>
<td>$S_{33}$ to $S_{35}$</td>
<td>See Table 3-21</td>
<td>Code rate, LP stream</td>
</tr>
</tbody>
</table>
Table 3-17. PS Signalling Information and Format (continued)

<table>
<thead>
<tr>
<th>Bits</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_{36} - S_{37}$</td>
<td>See Table 3-22 Guard interval</td>
</tr>
<tr>
<td>$S_{38} - S_{39}$</td>
<td>See Table 3-23 Transmission mode</td>
</tr>
<tr>
<td>$S_{40} - S_{52}$</td>
<td>all set to 0 Reserved for future use</td>
</tr>
<tr>
<td>$S_{54} - S_{66}$</td>
<td>BCH code Error protection</td>
</tr>
</tbody>
</table>

Table 3-18. Signalling Format for Frame Number

<table>
<thead>
<tr>
<th>Bits $S_{23} - S_{24}$</th>
<th>Frame Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>Frame number 1 in the super-frame</td>
</tr>
<tr>
<td>01</td>
<td>Frame number 2 in the super-frame</td>
</tr>
<tr>
<td>10</td>
<td>Frame number 3 in the super-frame</td>
</tr>
<tr>
<td>11</td>
<td>Frame number 4 in the super-frame</td>
</tr>
</tbody>
</table>

Table 3-19. Signalling Format for Possible Constellation Patterns

<table>
<thead>
<tr>
<th>Bits $S_{25} - S_{26}$</th>
<th>Constellation Characteristics</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>QPSK</td>
</tr>
<tr>
<td>01</td>
<td>16-QAM</td>
</tr>
<tr>
<td>10</td>
<td>64-QAM</td>
</tr>
<tr>
<td>11</td>
<td>reserved</td>
</tr>
</tbody>
</table>

Table 3-20. Signalling Format for $\alpha$ Values

<table>
<thead>
<tr>
<th>Bits $S_{27} - S_{29}$</th>
<th>$\alpha$ Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>000</td>
<td>Non-Hierarchical</td>
</tr>
<tr>
<td>001</td>
<td>$\alpha = 1$</td>
</tr>
<tr>
<td>010</td>
<td>$\alpha = 2$</td>
</tr>
<tr>
<td>011</td>
<td>$\alpha = 3$</td>
</tr>
<tr>
<td>100</td>
<td>reserved</td>
</tr>
<tr>
<td>101</td>
<td>reserved</td>
</tr>
<tr>
<td>110</td>
<td>reserved</td>
</tr>
<tr>
<td>111</td>
<td>reserved</td>
</tr>
</tbody>
</table>
DVB-T Components

Table 3-21. Signalling Format for Each Code Rate

<table>
<thead>
<tr>
<th>Bits S30 - S32 HP stream</th>
<th>Code Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>000</td>
<td>1/2</td>
</tr>
<tr>
<td>001</td>
<td>2/3</td>
</tr>
<tr>
<td>010</td>
<td>3/4</td>
</tr>
<tr>
<td>011</td>
<td>5/6</td>
</tr>
<tr>
<td>100</td>
<td>7/8</td>
</tr>
<tr>
<td>101</td>
<td>reserved</td>
</tr>
<tr>
<td>110</td>
<td>reserved</td>
</tr>
<tr>
<td>111</td>
<td>reserved</td>
</tr>
</tbody>
</table>

Table 3-22. Signalling Format for Each Guard Interval

<table>
<thead>
<tr>
<th>Bits S36 - S37</th>
<th>Guard Interval Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>1/32</td>
</tr>
<tr>
<td>01</td>
<td>1/16</td>
</tr>
<tr>
<td>10</td>
<td>1/8</td>
</tr>
<tr>
<td>11</td>
<td>1/4</td>
</tr>
</tbody>
</table>

Table 3-23. Signalling Format for Transmission Mode

<table>
<thead>
<tr>
<th>Bits S38 - S39</th>
<th>Transmission Mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>2k mode</td>
</tr>
<tr>
<td>01</td>
<td>8k mode</td>
</tr>
<tr>
<td>10</td>
<td>reserved</td>
</tr>
<tr>
<td>11</td>
<td>reserved</td>
</tr>
</tbody>
</table>

References

**DTV_DVBTPSDemod**

**Description**  
Transmission parameter signal differential demodulation

**Library**  
DTV, DVB-T

**Class**  
SDFDTV_DVBTPSDemod

**Parameters**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Default</th>
<th>Type</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Length</td>
<td>length of TPS bits per OFDM frame</td>
<td>68</td>
<td>int</td>
<td>68</td>
</tr>
</tbody>
</table>

**Pin Inputs**

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>input</td>
<td>TPS information (68 bits)</td>
<td>complex</td>
</tr>
</tbody>
</table>

**Pin Outputs**

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>output</td>
<td>demodulated TPS information</td>
<td>int</td>
</tr>
<tr>
<td>3</td>
<td>init</td>
<td>demodulated initial bit for TPS DBPSK demodulation</td>
<td>int</td>
</tr>
</tbody>
</table>

**Notes/Equations**

1. The model is used to perform DBPSK demodulation for 68 complex signals about the received transmission parameter signal.

2. A hard decision value is made on the real part of the complex input signal.

\[
B'[i] = \begin{cases} 
0 & \text{if } \text{Re}(x[i]) > 0 \\
1 & \text{if } \text{Re}(x[i]) < 0 
\end{cases}
\]
DVB-T Components

where

\[ 0 \leq i < 68. \]

Then set \( B_0 = B'[0] \). DBPSK demodulation is:

\[ B_i = B'[i] \oplus B'[i-1] \]

where

\[ i = 1, 2, ... , 67. \]

References

**DTV_DVBTPSMod**

**Description**  Transmission parameter signal differential modulation  
**Library**  DTV, DVB-T  
**Class**  SDFDTV_DVBTPSMod

### Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Default</th>
<th>Type</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Length</td>
<td>length of TPS bits per OFDM frame</td>
<td>68</td>
<td>int</td>
<td>68</td>
</tr>
</tbody>
</table>

### Pin Inputs

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>input</td>
<td>TPS information (67 bits)</td>
<td>int</td>
</tr>
<tr>
<td>2</td>
<td>initial</td>
<td>initial bit for TPS DBPSK modulation</td>
<td>int</td>
</tr>
</tbody>
</table>

### Pin Outputs

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>output</td>
<td>modulated TPS information</td>
<td>complex</td>
</tr>
</tbody>
</table>

### Notes/Equations

1. The model is used to perform DBPSK modulation for transmission parameter signal information.
DVB-T Components

2. Implementation
First, set, \( B'[0] = B_0 \). DBPSK modulation is
\[
B'[i] = B_i \oplus B'[i-1]
\]
where \( i = 1, 2, \ldots, 67 \).
Then
\[
\begin{align*}
x[i] & = 1 + j0 \quad \text{if } (B'[i] = 0), \\
x[i] & = -1 + j0 \quad \text{if } (B'[i] = 1)
\end{align*}
\]
coded bits \( B'[0] \sim B'[67] \) are converted to \((1,0), (-1, 0)\)
where
\( i = 0, 1, \ldots, 67 \).

References
**DTV_DVBTReceiver**

**Description**  
DVB-T receiver

**Library**  
DTV, DVB-T

**Required Licenses**

**Parameters**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Default</th>
<th>Sym</th>
<th>Type</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mode</td>
<td>transmission mode: DVB 2k mode, DVB 8k mode</td>
<td>DVB 2k mode</td>
<td>enum</td>
<td></td>
<td></td>
</tr>
<tr>
<td>OversamplingOption</td>
<td>oversampling ratio option: Ratio 1, Ratio 2, Ratio 4, Ratio 8, Ratio 16, Ratio 32</td>
<td>Ratio 1</td>
<td>S</td>
<td>enum</td>
<td></td>
</tr>
<tr>
<td>GuardInterval</td>
<td>guard interval (frac FFT size)</td>
<td>0.125</td>
<td>real</td>
<td>[0, 1]</td>
<td></td>
</tr>
<tr>
<td>CodeRate</td>
<td>code rate: DTV 1/2, DTV 2/3, DTV 3/4, DTV 5/6, DTV 7/8</td>
<td>DTV 1/2</td>
<td>enum</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MappingMode</td>
<td>signal constellations and mapping: QPSK, QAM-16, QAM-64</td>
<td>QAM-16</td>
<td>enum</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SoftDecision</td>
<td>soft decision viterbi decoding type: Soft, Hard, Cochannel</td>
<td>Soft</td>
<td>enum</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TrunLen</td>
<td>path memory truncation length of Viterbi decoding algorithm, in bytes</td>
<td>10</td>
<td>int</td>
<td>[5, ∞)</td>
<td></td>
</tr>
</tbody>
</table>

**Pin Inputs**

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Rx_signal</td>
<td>received OFDM signal to be demodulated</td>
<td>complex</td>
</tr>
</tbody>
</table>

**Pin Outputs**

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>Bytes_Decoded</td>
<td>decoded bytes after Reed Solomon decoder</td>
<td>int</td>
</tr>
<tr>
<td>3</td>
<td>Bytes_AfterViterbi</td>
<td>decoded bytes after Viterbi decoder</td>
<td>int</td>
</tr>
</tbody>
</table>
DVB-T Components

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>TPS_Bits</td>
<td>decoded TPS information bits</td>
<td>int</td>
</tr>
<tr>
<td>5</td>
<td>constellation</td>
<td>constellation signal after OFDM demodulation</td>
<td>complex</td>
</tr>
</tbody>
</table>

Notes/Equations

1. This subnetwork model implements the DVB-T receiver function. The schematic for this subnetwork is shown in Figure 3-31.

[Figure 3-31. DTV_DVBTReceiver Schematic]
2. Output Delays

The constellation pin has one OFDM symbol delay. The first $8 \times 1705$ (2k mode) or $8 \times 6817$ (8k mode) constellation output signals not used for EVM calculation.

DTV_BCHDecoder works a 68-bit block; so, a Delay component with $N=60$ is inserted before DTV_DVBTPSDemod. TPS_Bits has a one DVB-T frame (68 OFDM symbols) delay; the first 53 bits are not used.

The information bytes per one OFDM symbol in Bytes_AfterViterbi can be calculated as follows:

- DataPerOFDMSymbol = 1512 (2k mode) or 6048 (8k mode)
- CodedBlockSize = 204
- CodeRate = 1/2, 2/3, 3/4, 5/6 or 7/8,
- InfoBytes = DelayOFDMSymbols $\times$ DataPerOFDMSymbol $\times$ CodeRate / 8
- DelayOFDMSymbols = 8

The block delay of Bytes_AfterViterbi is BlockDelay_AfterViterbi and can be calculated, as follows:

- DelayBytes = DelayOFDMSymbols $\times$ InfoBytes - (int(DelayOFDMSymbols $\times$ InfoBytes/CodedBlockSize)) $\times$ CodedBlockSize,
- $N = \text{int}((\text{TrunLen} + \text{DelayBytes}) / \text{CodedBlockSize})$,
- BlockDelay_AfterViterbi = 1 + int((DelayOFDMSymbols $\times$ InfoBytes) / CodedBlockSize) + $N$.

The block size is 204 bytes, $N$ is used when the sum of TrunLen and DelayBytes is more than one block length.

The Bytes_Decoded pin has 11 extra transport MUX packet delays.

The block delay of Bytes_Decoded is

- BlockDelay_Decoded = 1 + int((DelayOFDMSymbols $\times$ InfoBytes) / CodedBlockSize) + $N$.

The block size is 188 bytes.

3. Receiver functions are implemented according to the DVB-T Standard.

Start of OFDM symbol is detected by the DTV_DVBT_OFDMDemod subnetwork (shown in Figure 3-32).
DTV_MLEstimator calculates the correlation based on guard interval; DTV_LoadFFTBuff selects the index with the maximum correlation value as the start of FFT.

DTV_DVBCh2DEstimator estimates the complex channel impulse responses for continual and scattered pilot subcarriers. The channel impulse responses of the data subcarriers are 2D interpolated based on the estimated CIR.

Each subcarrier value is divided with a complex estimated channel response coefficient (phase tracking, phase synchronization, and equalization). This simple one-tap frequency domain channel response compensation is implemented by DTV_OFDMEqualizer.

After equalization, DTV_DVBDemuxOFDMSym demultiplexes 1705 or 6817 subcarriers into 1512 (2k mode) or 6048 (8k mode) data subcarriers and 1 demodulated TPS.

The demodulated 1512 or 6048 data subcarriers (such as 16-QAM and 64-QAM modulation) are demapped by DTV_Demapper.

Demodulated soft bits are de-interleaved by DTV_DVBTInnerDeinterlv.
De-interleaved bits are then inner decoded (DTV_InnerDecoder), outer de-interleaved (DTV_InterlvInt), outer decoded (DTV_RSDecoder) and descrambled (DTV_ScrambleByte).

The fully decoded stream bytes are then output as Bytes_Decoded.

The partially decoded stream bytes after inner decoding are output as Bytes_AfterViterbi.

The TPS signal is demodulated by DTV_DVBTPSDemod; the demodulated bits are decoded by DTV_BCHDecoder.

The output pins of this DTV_DVBTRceiver subnetwork correspond to those of DTV_DVBTSignalSrc.

This receiver is triggered per one OFDM symbol. The number of samples of each OFDM symbol (Rx_signal input pin) is:

- \((1 + \text{GuardInterval}) \times 2048 \times \text{OversamplingRatio}\) for 2k mode
- \((1 + \text{GuardInterval}) \times 8192 \times \text{OversamplingRatio}\) for 8k mode

The number of signals at the constellation output pin is 1705 (2k mode) or 6817 (8k mode) per one OFDM symbol.

4. Parameter Details

Mode specifies the 2k or 8k transmission mode as defined in DVB-T.

OversamplingOption specifies the transmission signal oversampling ratio 1, 2, 4, 8, 16, 32. If OversamplingOption = Ratio 2, the IFFT size is 4096 (2k mode) or 16384 (8k mode).

GuardInterval specifies the guard interval (also called cyclic extension) length for each symbol time, as a fraction of the FFT time period. Guard interval types 1/4, 1/8, 1/16 and 1/32 are defined in the DVB-T specification. GuardInterval can be set to any value between 0.0 and 1.0 (not just 1/4, 1/8, 1/16 and 1/32). For proper demodulation, the value must match the guard interval length actually used in the input signal.

CodeRate specifies the DTV code rate: 1/2, 2/3, 3/4, 5/6 or 7/8.

MappingMode specifies signal constellation and mapping modes: QPSK, QAM-16, or QAM-64.

SoftDecision specifies the Viterbi decoding algorithm mode.
DVB-T Components

- If SoftDecision=Soft, the Viterbi decoding algorithm is a soft decision decoder and uses channel status information.
- If SoftDecision=Hard, the Viterbi decoding algorithm is a hard decision decoder and does not use channel status information.
- If SoftDecision=Cochannel, the Viterbi decoding algorithm is for co-channel measurement with PAL analog TV and digital DVB-T.

TrunLen sets the truncation length (in bytes) in the Viterbi decoding algorithm. The path memory is 8×TrunLen bits in Viterbi decoding algorithm.

References


**Description**  
DVB-T RF receiver

**Library**  
DTV, DVB-T

---

**Parameters**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Default</th>
<th>Sym</th>
<th>Unit</th>
<th>Type</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>RIn</td>
<td>input resistance</td>
<td>DefaultRIn</td>
<td>Ohm</td>
<td>real</td>
<td>(0, ∞)</td>
<td></td>
</tr>
<tr>
<td>ROut</td>
<td>output resistance</td>
<td>DefaultROut</td>
<td>Ohm</td>
<td>real</td>
<td>(0, ∞)</td>
<td></td>
</tr>
<tr>
<td>RTemp</td>
<td>physical temperature, in degrees C</td>
<td>DefaultRTemp</td>
<td>Celsius</td>
<td>real</td>
<td>(-273.15, ∞)</td>
<td></td>
</tr>
<tr>
<td>GainImbalance</td>
<td>Gain imbalance, Q vs I</td>
<td>0.0</td>
<td>dB</td>
<td>real</td>
<td>(-∞, ∞)</td>
<td></td>
</tr>
<tr>
<td>PhaseImbalance</td>
<td>Phase imbalance, Q vs I</td>
<td>0.0</td>
<td>deg</td>
<td>real</td>
<td>(-∞, ∞)</td>
<td></td>
</tr>
<tr>
<td>FCarrier</td>
<td>carrier frequency</td>
<td>474.0MHz</td>
<td>Hz</td>
<td>real</td>
<td>(0, ∞)</td>
<td></td>
</tr>
<tr>
<td>Sensitivity</td>
<td>voltage output sensitivity, Vout/Vin</td>
<td>1</td>
<td>real</td>
<td></td>
<td>(-∞, ∞)</td>
<td></td>
</tr>
<tr>
<td>Phase</td>
<td>reference phase in degrees</td>
<td>0.0</td>
<td>deg</td>
<td>real</td>
<td>(-∞, ∞)</td>
<td></td>
</tr>
<tr>
<td>Mode</td>
<td>transmission mode: DVB 2k mode, DVB 8k mode</td>
<td>DVB 2k mode</td>
<td>enum</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OversamplingOption</td>
<td>oversampling ratio option: Ratio 1, Ratio 2, Ratio 4, Ratio 8, Ratio 16, Ratio 32</td>
<td>Ratio 1</td>
<td>S</td>
<td>enum</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GuardInterval</td>
<td>guard interval (frac FFT size)</td>
<td>0.125</td>
<td>real</td>
<td></td>
<td>[0, 1]</td>
<td></td>
</tr>
<tr>
<td>CodeRate</td>
<td>code rate: DTV 1/2, DTV 2/3, DTV 3/4, DTV 5/6, DTV 7/8</td>
<td>DTV 1/2</td>
<td>enum</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MappingMode</td>
<td>signal consellations and mapping: QPSK, QAM-16, QAM-64</td>
<td>QAM-16</td>
<td>enum</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SoftDecision</td>
<td>soft decision viterbi decoding type: Soft, Hard, Cochannel</td>
<td>Soft</td>
<td>enum</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TruncLen</td>
<td>path memory truncation length of Viterbi decoding algorithm, in bytes</td>
<td>10</td>
<td>int</td>
<td></td>
<td>[5, ∞)</td>
<td></td>
</tr>
</tbody>
</table>
DVB-T Components

Pin Inputs

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Rx_signal</td>
<td>received OFDM signal to be demodulated</td>
<td>timed</td>
</tr>
</tbody>
</table>

Pin Outputs

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>Bytes_Decoded</td>
<td>decoded bytes after Reed Solomon decoder</td>
<td>int</td>
</tr>
<tr>
<td>3</td>
<td>Bytes_AfterViterbi</td>
<td>decoded bytes after Viterbi decoder</td>
<td>int</td>
</tr>
<tr>
<td>4</td>
<td>TPS_Bits</td>
<td>decoded TPS information bits</td>
<td>int</td>
</tr>
<tr>
<td>5</td>
<td>constellation</td>
<td>constellation signal after OFDM demodulation</td>
<td>complex</td>
</tr>
</tbody>
</table>

Notes/Equations

1. This subnetwork model implements the DVB-T RF receiver function. The received RF signal is demodulated and fed to the baseband receiver.

2. The schematic for this RF receiver is shown in Figure 3-33.

3. The baseband demodulator schematic is shown in Figure 3-34.
4. Output Delays

The constellation pin has one OFDM symbol delay. The first $8 \times 1705$ (2k mode) or $8 \times 6817$ (8k mode) constellation output signals not used for EVM calculation.

DTV_BCHDecoder works a 68-bit block; so, a Delay component with $N=60$ is inserted before DTV_DVBTPSDemod. TPS_Bits has a one DVB-T frame (68 OFDM symbols) delay; the first 53 bits are not used.

The information bytes for one OFDM symbol in Bytes_AfterViterbi can be calculated as follows:

- $\text{DataPerOFDMSymbol} = 1512$ (2k mode) or 6048 (8k mode)
- $\text{CodedBlockSize} = 204$
- $\text{CodeRate} = 1/2, 2/3, 3/4, 5/6$ or $7/8$,
- $\text{InfoBytes} = \text{DelayOFDM_symbols} \times \text{DataPerOFDMSymbol} \times \text{CodeRate}/8$
- $\text{DelayOFDMSymbols} = 8$

The block delay of Bytes_AfterViterbi is BlockDelay_AfterViterbi and can be calculated, as follows:
DVB-T Components

- DelayBytes = DelayOFDM Symbols × InfoBytes - (int(DelayOFDM Symbols × InfoBytes/CodedBlockSize)) × CodedBlockSize,
- \( N = \text{int}((\text{TrunLen} + \text{DelayBytes})/\text{CodedBlockSize}), \)
- \( \text{BlockDelay\_AfterViterbi} = 1 + \text{int}((\text{DelayOFDM Symbols} \times \text{InfoBytes})/\text{CodedBlockSize}) + N. \)

The block size is 204 bytes, \( N \) is used when the sum of \( \text{TrunLen} \) and \( \text{DelayBytes} \) is more than one block length.

The block size is 204 bytes, \( N \) is used when the sum of \( \text{TrunLen} \) and \( \text{DelayBytes} \) is more than one block length.

The Bytes_Decoded pin has 11 extra transport MUX packet delays.

The block delay of Bytes_Decoded is
- \( \text{BlockDelay\_Decoded} = 1 + \text{int}((\text{DelayOFDM Symbols} \times \text{InfoBytes})/\text{CodedBlockSize}) + N. \)

The block size is 188 bytes.

5. Receiver functions are implemented according to the DVB-T Standard.

Start of OFDM symbol is detected by the DTV_DVBTOFDMDemod subnetwork (shown in Figure 3-35).
DTV_MLEstimator calculates the correlation based on guard interval; DTV_LoadFFTBuff selects the index with the maximum correlation value as the start of FFT.

DTV_DVBCh2DEstimator estimates the complex channel impulse responses for continual and scattered pilot subcarriers. The channel impulse responses of the data subcarriers are 2D interpolated based on the estimated CIR.

Each subcarrier value is divided with a complex estimated channel response coefficient (phase tracking, phase synchronization, and equalization). This simple one-tap frequency domain channel response compensation is implemented by DTV_OFDMEqualizer.

After equalization, DTV_DVBDemuxOFDMSym demultiplexes 1705 or 6817 subcarriers into 1512 (2k mode) or 6048 (8k mode) data subcarriers and 1 demodulated TPS.

The demodulated 1512 or 6048 data subcarriers (16-QAM and 64-QAM modulation) are demapped by DTV_Demapper.

Demodulated soft bits are de-interleaved by DTV_DVBTInnerDeinterlv.
De-interleaved bits are then inner decoded (DTV_InnerDecoder), outer de-interleaved (DTV_Intervlnt), outer decoded (DTV_RSPDecoder) and descrambled (DTV_ScrambleByte).

The fully decoded stream bytes are then output as Bytes_Decoded.

The partially decoded stream bytes after inner decoding are output as Bytes_AfterViterbi.

The TPS signal is demodulated by DTV_DVTPSDemod; the demodulated bits are decoded by DTV_BCHDecoder.

The output pins of this DTV_DVBTReceiver subnetwork correspond to those of DTV_DVBTSignalSrc.

This receiver is triggered at each OFDM symbol. The number of samples of each OFDM symbol (Rx_signal input pin) is:

- \((1+\text{GuardInterval})\times2048\times\text{OversamplingRatio}\) for 2k mode
- \((1+\text{GuardInterval})\times8192\times\text{OversamplingRatio}\) for 8k mode

The number of signals at the constellation output pin is 1705 (2k mode) or 6817 (8k mode) for one OFDM symbol.

6. Parameter Details

GainImbalance and PhaseImbalance add certain impairments to the ideal output RF signal. Impairments are added in the order described here.

The unimpaired RF I and Q envelope voltages have gain and phase imbalance applied. The RF is given by:

\[
V_3(t) = A\bigg(V_1(t)\cos(\omega_c t) - gV_2(t)\sin(\omega_c t + \frac{\phi\pi}{180})\bigg)
\]

where \(V_3(t)\) is the in-phase RF envelope, \(V_2(t)\) is the quadrature phase RF envelope, \(g\) is the gain imbalance

\[
g = \frac{\text{GainImbalance}}{10} \times 10
\]

and, \(\phi\) (in degrees) is the phase imbalance.

Mode specifies the 2k or 8k transmission mode as defined in DVB-T.
OversamplingOption specifies the transmission signal oversampling ratio 1, 2, 4, 8, 16, 32. If OversamplingOption=Ratio 2, the IFFT size is 4096 for 2k mode or 16384 for 8k mode.

GuardInterval specifies the guard interval (also called cyclic extension) length for each symbol time, as a fraction of the FFT time period. Guard interval types 1/4, 1/8, 1/16 and 1/32 are defined in the DVB-T specification. GuardInterval can be set to any value between 0.0 and 1.0 (not just 1/4, 1/8, 1/16 and 1/32). For proper demodulation, the value must match the guard interval length actually used in the input signal.

CodeRate specifies the stream code rate: 1/2, 2/3, 3/4, 5/6, or 7/8.

MappingMode specifies signal constellations and mapping for the DVB-T signal source. Mapping modes QPSK, 16-QAM and 64-QAM are available.

SoftDecision specifies the Viterbi decoding algorithm mode.
- If SoftDecision=Soft, the Viterbi decoding algorithm is a soft decision decoder and uses channel status information.
- If SoftDecision=Hard, the Viterbi decoding algorithm is a hard decision decoder and does not use channel status information.
- If SoftDecision=Cochannel, the Viterbi decoding algorithm is for co-channel measurement with PAL analog TV and digital DVB-T.

TrunLen is set to the truncation length in the Viterbi decoding algorithm. This parameter is set as bytes. The path memory is 8×TrunLen bits in Viterbi decoding algorithm.

TrunLen sets the truncation length (in bytes) in the Viterbi decoding algorithm. The path memory is 8×TrunLen bits in Viterbi decoding algorithm.

References
[1] ETSI, Digital Video Broadcasting (DVB); Framing structure, channel coding and modulation for digital terrestrial television. EN 300 744 v1.2.1, European Telecommunication Standard, July 1999
DVB-T Components

**DTV_DVBTSignalSrc**

**Description**  
DVB-T signal source

**Library**  
DTV, DVB-T

**Parameters**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Default</th>
<th>Sym</th>
<th>Type</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mode</td>
<td>transmission mode: DVB 2k mode, DVB 8k mode</td>
<td>DVB 2k mode</td>
<td></td>
<td>enum</td>
<td></td>
</tr>
<tr>
<td>OversamplingOption</td>
<td>oversampling ratio option: Ratio 1, Ratio 2, Ratio 4, Ratio 8, Ratio 16, Ratio 32</td>
<td>Ratio 1</td>
<td>S</td>
<td>enum</td>
<td></td>
</tr>
<tr>
<td>GuardInterval</td>
<td>guard interval (fractional FFT size)</td>
<td>0.125</td>
<td>real</td>
<td>[0, 1]</td>
<td></td>
</tr>
<tr>
<td>CodeRate</td>
<td>code rate: DTV 1/2, DTV 2/3, DTV 3/4, DTV 5/6, DTV 7/8</td>
<td>DTV 1/2</td>
<td></td>
<td>enum</td>
<td></td>
</tr>
<tr>
<td>MappingMode</td>
<td>signal constellation and mapping: QPSK, QAM-16, QAM-64</td>
<td>QAM-16</td>
<td></td>
<td>enum</td>
<td></td>
</tr>
<tr>
<td>DataType</td>
<td>payload data type: PN9, PN15, FIX4, _4_1_4_0, _8_1_8_0, _16_1_16_0, _32_1_32_0, _64_1_64_0</td>
<td>PN9</td>
<td></td>
<td>enum</td>
<td></td>
</tr>
</tbody>
</table>

**Pin Outputs**

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>DVB_T_signal</td>
<td>DVB-T signal</td>
<td>complex</td>
</tr>
<tr>
<td>2</td>
<td>Bytes_Uncoded</td>
<td>information bytes of transport packet</td>
<td>int</td>
</tr>
<tr>
<td>3</td>
<td>Bytes_BeforeCC</td>
<td>information bytes before convolutional coder</td>
<td>int</td>
</tr>
<tr>
<td>4</td>
<td>TPS_Bits</td>
<td>TPS information bits</td>
<td>int</td>
</tr>
<tr>
<td>5</td>
<td>constellation</td>
<td>constellation signal before IFFT</td>
<td>complex</td>
</tr>
</tbody>
</table>

**Notes/Equations**

3-114  DTV_DVBTSignalSrc
1. This subnetwork model is used to generate a DVB-T signal source. One OFDM symbol is generated each firing. The schematic for this subnetwork is shown in Figure 3-36.

2. The DVB-T signal source format follows the DVB-T specification. Figure 3-37 is functional block diagram of the DVB-T baseband system. The DTV_DVBSignalSrc includes: scrambler (DTV_ScrambleByte); outer coder (DTV_RSCoder); outer interleaver (DTV_InterlvInt); inner coder (DTV_PuncConvCoder); inner interleaver (DTV_DVBTInnerInterlv); mapper (DTV_Mapper); and, frame adaptation, OFDM, and guard interval insertion (in DTV_DVBT_OFDMMod subnetwork).
DVB-T Components

For a MPEG-2 transport stream, the DTV_DVBTSSource model can be activated (shown de-activated in Figure 3-36) and the current active source deactivated. DTV_DVBTSignalSrc requires more buffer than the current active source; for example, a code rate of 7/8 is too large.

This signal source works per one OFDM symbol. The number of constellation output pin is 1705 (2k mode) or 6817 (8k mode) per one OFDM symbol. The number of samples of each OFDM symbol (DVBT_signal output pin) is:

- \[(1+\text{GuardInterval}) \times 2048 \times \text{OversamplingRatio} \text{ for 2k mode;}
- \[(1+\text{GuardInterval}) \times 8192 \times \text{OversamplingRatio} \text{ for 8k mode.}

3. Parameter Details

Mode is used to select a 2k or 8k transmission mode as defined in DVB-T.

OversamplingOption indicates the oversampling ratio of the transmission signal. Oversampling ratios 1, 2, 4, 8, 16, 32 are supported in this source. If OversamplingOption = Ratio 2, the IFFT size is 4096 (2k mode) or 16384 (8k mode).

GuardInterval specifies the guard interval (also called cyclic extension) length for each symbol time, as a fraction of the FFT time period. Guard interval types 1/4, 1/8, 1/16 and 1/32 are defined in the DVB-T specification. GuardInterval can be set to any value between 0.0 and 1.0 (not just 1/4, 1/8, 1/16 and 1/32). For proper demodulation, the value must match the guard interval length actually used in the input signal.

CodeRate is defined as code rate of stream. Code rates 1/2, 2/3, 3/4, 5/6 and 7/8 are available in this source.

MappingMode specifies signal constellations and mapping: QPSK, 16-QAM or 64-QAM.

For the DataType parameter:
• if PN9 is selected, a 511-bit pseudo-random test pattern is generated according to CCITT Recommendation O.153
• if PN15 is selected, a 32767-bit pseudo-random test pattern is generated according to CCITT Recommendation O.151
• if FIX4 is selected, a zero-stream is generated
• if x_1_x_0 is selected (x equals 4, 8, 16, 32, or 64) a periodic bit stream is generated, with the period being 2x. In one period, the first x bits are 1s and the second x bits are 0s.

References
# DVB-T Components

## DTV_DVBTSignalSrc_RF

### Description
DVB-T RF signal source

### Library
DTV, DVB-T

### Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Default</th>
<th>Sym</th>
<th>Unit</th>
<th>Type</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>ROut</td>
<td>output resistance</td>
<td>DefaultROut</td>
<td>Ohm</td>
<td>real</td>
<td>(0, ∞)</td>
<td></td>
</tr>
<tr>
<td>RTemp</td>
<td>physical temperature, in degrees C</td>
<td>DefaultRTemp</td>
<td>Celsius</td>
<td>real</td>
<td>[-273.15, ∞)</td>
<td></td>
</tr>
<tr>
<td>Noise</td>
<td>Enable thermal noise? NO, YES</td>
<td>NO</td>
<td>enum</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FCarrier</td>
<td>carrier frequency</td>
<td>474.0MHz</td>
<td>Hz</td>
<td>real</td>
<td>(0, ∞)</td>
<td></td>
</tr>
<tr>
<td>Power</td>
<td>modulator output power</td>
<td>40mW</td>
<td>W</td>
<td>real</td>
<td>(0, ∞)</td>
<td></td>
</tr>
<tr>
<td>Bandwidth</td>
<td>bandwidth</td>
<td>(2048/224.0)MHz</td>
<td>Hz</td>
<td>real</td>
<td>(0, ∞)</td>
<td></td>
</tr>
<tr>
<td>MirrorSpectrum</td>
<td>Mirror spectrum about carrier? NO, YES</td>
<td>NO</td>
<td>enum</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GainImbalance</td>
<td>Gain imbalance, Q vs I</td>
<td>0.0</td>
<td>dB</td>
<td>real</td>
<td>(-∞, ∞)</td>
<td></td>
</tr>
<tr>
<td>PhaseImbalance</td>
<td>Phase imbalance, Q vs I</td>
<td>0.0</td>
<td>deg</td>
<td>real</td>
<td>(-∞, ∞)</td>
<td></td>
</tr>
<tr>
<td>I_OriginOffset</td>
<td>I origin offset (percent)</td>
<td>0.0</td>
<td>real</td>
<td></td>
<td>(-∞, ∞)</td>
<td></td>
</tr>
<tr>
<td>Q_OriginOffset</td>
<td>Q origin offset (percent)</td>
<td>0.0</td>
<td>real</td>
<td></td>
<td>(-∞, ∞)</td>
<td></td>
</tr>
<tr>
<td>IQ_Rotation</td>
<td>IQ rotation</td>
<td>0.0</td>
<td>deg</td>
<td>real</td>
<td>(-∞, ∞)</td>
<td></td>
</tr>
<tr>
<td>Mode</td>
<td>transmission mode: DVB 2k mode, DVB 8k mode</td>
<td>DVB 2k mode</td>
<td></td>
<td>enum</td>
<td></td>
<td></td>
</tr>
<tr>
<td>OversamplingOption</td>
<td>oversampling ratio option: Ratio 1, Ratio 2, Ratio 4, Ratio 8, Ratio 16, Ratio 32</td>
<td>Ratio 1</td>
<td>S</td>
<td>enum</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GuardInterval</td>
<td>guard interval (fractional FFT size)</td>
<td>0.125</td>
<td>real</td>
<td></td>
<td>[0, 1]</td>
<td></td>
</tr>
<tr>
<td>CodeRate</td>
<td>code rate: DTV 1/2, DTV 2/3, DTV 3/4, DTV 5/6, DTV 7/8</td>
<td>DTV 1/2</td>
<td></td>
<td>enum</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
This subnetwork model integrates an RF modulator and a baseband signal source. The baseband signal is fed to the RF modulator; after RF modulation the timed signal is output.

The schematic for this subnetwork is shown in Figure 3-38.
DVB-T Components

2. The format of DVB-T signal source follows the DVB-T specification. Figure 3-39 is a functional block diagram of a DVB-T baseband system.

3. The schematic for the signal source is shown in Figure 3-40. The DTV_DVBTSignalSrc includes: scrambler (DTV_ScrambleByte); outer coder (DTV_RSCoder); outer interleaver (DTV_InterlvInt); inner coder (DTV_PuncConvCoder); inner interleaver (DTV_DVBTInnerInterlv); mapper (DTV_Mapper); and, frame adaptation, OFDM, and guard interval insertion (in DTV_DVBTOFDMMod subnetwork).

**Note** To use an MPEG-2 transport stream, DTV_DVBTTSPSource must be active and the current active source deactivated. DTV_DVBTTSPSource needs more buffer than the current active source. In some cases, code rate 7/8 is too large.
This signal source uses one OFDM symbol; 1705 (2k mode) or 6817 (8k mode) constellations are output. DVBT_signal pin outputs samples for each OFDM symbol:

- \((1+\text{GuardInterval}) \times 2048 \times \text{OversamplingRatio}\) for 2k mode
- \((1+\text{GuardInterval}) \times 8192 \times \text{OversamplingRatio}\) for 8k mode

4. Parameter Details

- **FCarrier** defines the RF frequency for the DVB signal.
- **Power** defines the power level for FCarrier.
- **Bandwidth** indicates signal bandwidth.
- The **MirrorSpectrum** (when set to YES) conjugates the input signal before any other processing is done.

The **GainImbalance**, **PhaseImbalance**, **I_OriginOffset**, **Q_OriginOffset**, and **IQ_Rotation** parameters add certain impairments to the ideal output RF signal. Impairments are added in the order described here.

The unimpaired RF I and Q envelope voltages have gain and phase imbalance applied. The RF is given by:
DVB-T Components

\[ V_3(t) = A \left( V_1(t) \cos(\omega_c t) - gV_2(t) \sin\left(\omega_c t + \frac{\phi \pi}{180}\right)\right) \]

where \( A \) is a scaling factor that depends on the Power and ROut parameters specified by the user, \( V_1(t) \) is the in-phase RF envelope, \( V_2(t) \) is the quadrature phase RF envelope, \( g \) is the gain imbalance, and \( \phi \) (in degrees) is the phase imbalance.

The signal \( V_3(t) \) is rotated by IQ_Rotation degrees; I_OriginOffset and Q_OriginOffset are then applied to the rotated signal. Note that the amounts specified are percentages with respect to the output rms voltage. The output rms voltage is given by \( \sqrt{2 \times ROut \times \text{Power}} \).

Mode specifies the 2k or 8k transmission mode as defined in the DVB-T specification.

OversamplingOption specifies the transmission signal oversampling ratio 1, 2, 4, 8, 16, 32. If OversamplingOption=2, the IFFT size is 4096 (2k mode) or 16384 (8k mode).

GuardInterval specifies the guard interval (also called cyclic extension) length for each symbol time, as a fraction of the FFT time period. Guard interval types 1/4, 1/8, 1/16 and 1/32 are defined in the DVB-T specification. GuardInterval can be set to any value between 0.0 and 1.0 (not just 1/4, 1/8, 1/16 and 1/32). For proper demodulation, the value must match the guard interval length actually used in the input signal.

CodeRate specifies the stream code rate: 1/2, 2/3, 3/4, 5/6, or 7/8.

MappingMode specifies signal constellation and mapping modes: QPSK, 16-QAM, or 64-QAM.

For the DataType parameter:
- if PN9 is selected, a 511-bit pseudo-random test pattern is generated according to CCITT Recommendation 0.153
- if PN15 is selected, a 32767-bit pseudo-random test pattern is generated according to CCITT Recommendation 0.151
- if FIX4 is selected, a zero-stream is generated
if \( x_1 \times x_0 \) is selected, where \( x \) equals 4, 8, 16, 32, or 64, a periodic bit stream is generated, with the period being 2\( x \). In one period, the first \( x \) bits are 1s and the second \( x \) bits are 0s.

References

### DVB-T Components

**DTV_DVBTTPSGen**

**Description**  
Modulated transmission parameter signal information

**Library**  
DTV, DVB-T

**Required Licenses**

#### Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Default</th>
<th>Type</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Length</td>
<td>length of TPS information bits</td>
<td>54</td>
<td>int</td>
<td>(54)</td>
</tr>
<tr>
<td>InitBit</td>
<td>initialization bit for DBPSK modulation (1 bit)</td>
<td>1</td>
<td>int</td>
<td>[0, 1]</td>
</tr>
<tr>
<td>Sync</td>
<td>synchronization word (16 bits): W0, W1</td>
<td>W0</td>
<td>enum</td>
<td></td>
</tr>
<tr>
<td>LengthIndicator</td>
<td>TPS length indicator (6 bits)</td>
<td>0 1 0 1 1 1</td>
<td>int array</td>
<td>&quot;0 1 0 1 1 1&quot;</td>
</tr>
<tr>
<td>FrameNumber</td>
<td>frame number in one super frame (2 bits): F1, F2, F3, F4</td>
<td>F1</td>
<td>enum</td>
<td></td>
</tr>
<tr>
<td>Constellation</td>
<td>modulation scheme in DVB-T (2bits): QPSK, QAM 16, QAM 64, reserved M</td>
<td>QPSK</td>
<td>enum</td>
<td></td>
</tr>
<tr>
<td>HierarchyInform</td>
<td>signaling format for alpha values (3 bits): Non hierarchical, alpha 1, alpha 2, alpha 4, reserved H1, reserved H2, reserved H3, reserved H4</td>
<td>Non hierarchical</td>
<td>enum</td>
<td></td>
</tr>
<tr>
<td>CodeRateHP</td>
<td>high priority stream current code rate (3 bits): HP 1/2, HP 2/3, HP 3/4, HP 5/6, HP 7/8, Reserved C1 HP, Reserved C2 HP, Reserved C3 HP</td>
<td>HP 1/2</td>
<td>enum</td>
<td></td>
</tr>
<tr>
<td>CodeRateLP</td>
<td>low priority stream current code rate (3 bits): LP 1/2, LP 2/3, LP 3/4, LP 5/6, LP 7/8, Reserved C1 LP, Reserved C2 LP, Reserved C3 LP</td>
<td>LP 1/2</td>
<td>enum</td>
<td></td>
</tr>
</tbody>
</table>
1. This subnetwork model is used to generate modulated transmission parameter signal information used in TPS carriers in OFDM symbols. The schematic for this subnetwork is shown in Figure 3-41.

![Figure 3-41. DTV_DVBTTPSGen Schematic](image)

2. DTV_DVBTPS generates 54 bits (1 initialization, 16 synchronization, and 37 information); the 53 TPS synchronization and information transmission mode are extended with 14 parity bits of the BCH (67,53, t = 2) shortened code, derived from the original systematic BCH (127,113, t = 2) code, which is implemented by DTV_BCHCoder; the 68-bit TPS block is modulated by DBPSK, which is implemented by DTV_DVBTPSMod.

Each firing, the TPS_Bits pin outputs 53 bits (16 synchronization and 37 information) and TPS outputs 68 modulated TPS signals.

3. The TPS is defined over 68 consecutive OFDM symbols (one OFDM frame). Four consecutive frames correspond to one OFDM super-frame.
DVB-T Components

The reference sequence corresponding to the TPS carriers of the first symbol of each OFDM frame is used to initialize the TPS modulation on each TPS carrier. Each OFDM symbol conveys one TPS bit. Each TPS block (corresponding to one OFDM frame) contains 68 bits:

• 1 initialization bit
• 16 synchronization bits
• 37 information bits
• 14 redundancy bits for error protection

Of the 37 information bits, 31 are used (6 bits are reserved for future use and are set to zero).

4. Bit assignments are listed in Table 3-24.

<table>
<thead>
<tr>
<th>Bit Number</th>
<th>Format</th>
<th>Purpose/Content</th>
</tr>
</thead>
<tbody>
<tr>
<td>S0</td>
<td></td>
<td>Initialization bit for differential 2_PSK modulation</td>
</tr>
<tr>
<td>S1 - S16</td>
<td>0011010111101110 or 1100101000010001</td>
<td>Synchronization word</td>
</tr>
<tr>
<td>S17 - S22</td>
<td>010 111</td>
<td>Length indicator</td>
</tr>
<tr>
<td>S23 - S24</td>
<td>See Table 3-25</td>
<td>Frame number</td>
</tr>
<tr>
<td>S25 - S26</td>
<td>See Table 3-26</td>
<td>Constellation</td>
</tr>
<tr>
<td>S27 - S29</td>
<td>See Table 3-20</td>
<td>Hierarchy information</td>
</tr>
<tr>
<td>S30 - S32</td>
<td>See Table 3-21</td>
<td>Code rate, HP stream</td>
</tr>
<tr>
<td>S33 - S35</td>
<td>See Table 3-21</td>
<td>Code rate, LP stream</td>
</tr>
<tr>
<td>S36 - S37</td>
<td>See Table 3-22</td>
<td>Guard interval</td>
</tr>
<tr>
<td>S38 - S39</td>
<td>See Table 3-23</td>
<td>Transmission mode</td>
</tr>
<tr>
<td>S40 - S52</td>
<td>all set to 0</td>
<td>Reserved for future use</td>
</tr>
<tr>
<td>S54 - S66</td>
<td>BCH code</td>
<td>Error protection</td>
</tr>
</tbody>
</table>

Table 3-24. TPS Signal Information
Table 3-25. Signal Format for Frame Number

<table>
<thead>
<tr>
<th>Bits $S_{23} - S_{24}$</th>
<th>Frame Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>Frame number 1 in the super-frame</td>
</tr>
<tr>
<td>01</td>
<td>Frame number 2 in the super-frame</td>
</tr>
<tr>
<td>10</td>
<td>Frame number 3 in the super-frame</td>
</tr>
<tr>
<td>11</td>
<td>Frame number 4 in the super-frame</td>
</tr>
</tbody>
</table>

Table 3-26. Signal Format for Constellation Patterns

<table>
<thead>
<tr>
<th>Bits $S_{25} - S_{26}$</th>
<th>Constellation Characteristics</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>QPSK</td>
</tr>
<tr>
<td>01</td>
<td>16-QAM</td>
</tr>
<tr>
<td>10</td>
<td>64-QAM</td>
</tr>
<tr>
<td>11</td>
<td>reserved</td>
</tr>
</tbody>
</table>

Table 3-27. Signal Format for $\alpha$ Values

<table>
<thead>
<tr>
<th>Bits $S_{27} - S_{29}$</th>
<th>$\alpha$ Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>000</td>
<td>Non-hierarchical</td>
</tr>
<tr>
<td>001</td>
<td>$\alpha = 1$</td>
</tr>
<tr>
<td>010</td>
<td>$\alpha = 2$</td>
</tr>
<tr>
<td>011</td>
<td>$\alpha = 3$</td>
</tr>
<tr>
<td>100</td>
<td>reserved</td>
</tr>
<tr>
<td>101</td>
<td>reserved</td>
</tr>
<tr>
<td>110</td>
<td>reserved</td>
</tr>
<tr>
<td>111</td>
<td>reserved</td>
</tr>
</tbody>
</table>

Table 3-28. Signal Format for Each Code Rate

<table>
<thead>
<tr>
<th>Bits $S_{30} - S_{32}$ $S_{33} - S_{35}$</th>
<th>Code rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>HP stream</td>
<td></td>
</tr>
<tr>
<td>000</td>
<td>1/2</td>
</tr>
<tr>
<td>001</td>
<td>2/3</td>
</tr>
<tr>
<td>010</td>
<td>3/4</td>
</tr>
</tbody>
</table>
5. Error Protection of TPS

The 53 bits containing the TPS synchronization and information (bits \( s_1 \) - \( s_{53} \)) are extended with 14 parity bits of the BCH (67,53, \( t = 2 \)) shortened code, derived from the original systematic BCH (127,113, \( t = 2 \)) code.

The code generator polynomial is:

\[
h(x) = x^{14} + x^9 + x^8 + x^6 + x^5 + x^4 + x^2 + x + 1
\]

The shortened BCH code is implemented by adding 60 bits, all set to zero, before the information bits input of an BCH(127,113, \( t = 2 \)) encoder. After BCH encoding these null bits are discarded, leading to a BCH code word of 67 bits.

6. TPS Modulation
TPS cells are transmitted at the normal power level, that is, they are transmitted with energy equal to that of the mean of all data cells, $E[c \times c^*] = 1$.

Each TPS carrier is DBPSK modulated and conveys the same message. The DBPSK is initialized at the beginning of each TPS block.

The following rule applies for the differential modulation of carrier $k$ of symbol $l$ ($l > 0$) in frame $m$:

- if $s_1 = 0$, then
  \[
  \text{Re}\{c_{m,l,k}\} = \text{Re}\{c_{m,l-1,k}\}; \text{Im}\{c_{m,l,k}\} = 0
  \]
- if $s_1 = 1$, then
  \[
  \text{Re}\{c_{m,l,k}\} = -\text{Re}\{c_{m,l-1,k}\}; \text{Im}\{c_{m,l,k}\} = 0
  \]

The absolute modulation of the TPS carriers in the first symbol in a frame is derived from the reference sequence $w_k$ as follows:

\[
\text{Re}\{c_{m,1,k}\} = 2\left(\frac{1}{2} - w_k\right)
\]

\[
\text{Im}\{c_{m,1,k}\} = 0
\]

References

DVB-T Components

**DTV_DVBTTSPsource**

**Description**  
Transport packets signal source

**Library**  
DTV, DVB-T

**Required Licenses**

**Parameters**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Default</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>DataType</td>
<td>payload data type: PN9, PN15, FIX4, _4_1_4_0, _8_1_8_0, _16_1_16_0, _32_1_32_0, _64_1_64_0</td>
<td>PN9</td>
<td>enum</td>
</tr>
</tbody>
</table>

**Pin Outputs**

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>TSP</td>
<td>Transport system packet signal source</td>
<td>int</td>
</tr>
</tbody>
</table>

**Notes/Equations**

1. This subnetwork model is used to generate MPEG-2 transport MUX packets.
   The schematic for this subnetwork is shown in Figure 3-42.
2. Each firing, 8 transport packets with 8×188 bytes are generated as illustrated in Figure 3-43.

To provide an initialization signal for the descrambler, the MPEG-2 sync byte of the first transport packet in a group of 8 packets is bit-wise inverted from $47_{HEX}(71)$ to $B8_{HEX}(184)$. So, the first sync of the first packet is $B8_{HEX}(184)$ and the left 7 sync of the left 7 packets are all $47_{HEX}(71)$.

3. DataPattern specifies the type of data pattern:
   - PN9 specifies a 511-bit pseudo-random test pattern is generated according to CCITT Recommendation O.153
   - PN15 specifies a 32767-bit pseudo-random test pattern is generated according to CCITT Recommendation O.151
   - FIX4 specifies a zero-stream is generated

---

DTV_DVBTTSPsource 3-131
DVB-T Components

• $x_{1\cdot x\cdot 0}$ (where $x$ equals 4, 8, 16, 32, or 64) specifies a periodic bit stream is generated, with the period being $2 \times x$. In one period, the first $x$ bits are 1 and the second $x$ bits are 0.

References

DTV_PALSource

**Description**  
PAL signal source

**Library**  
DTV, DVB-T

**Required Licenses**

**Parameters**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Default</th>
<th>Unit</th>
<th>Type</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>RIn</td>
<td>output resistance</td>
<td>DefaultRIn</td>
<td>Ohm</td>
<td>real</td>
<td>(0, ∞)</td>
</tr>
<tr>
<td>ROut</td>
<td>output resistance</td>
<td>DefaultROut</td>
<td>Ohm</td>
<td>real</td>
<td>(0, ∞)</td>
</tr>
<tr>
<td>RTemp</td>
<td>physical temperature, in degrees C</td>
<td>DefaultRTemp</td>
<td>Celsius</td>
<td>real</td>
<td>(-273.15, ∞)</td>
</tr>
<tr>
<td>FCarrier</td>
<td>carrier frequency</td>
<td>474.0MHz</td>
<td>Hz</td>
<td>real</td>
<td>(0, ∞)</td>
</tr>
<tr>
<td>Power</td>
<td>modulator output power</td>
<td>40mW</td>
<td>W</td>
<td>real</td>
<td>(0, ∞)</td>
</tr>
<tr>
<td>VRef</td>
<td>reference voltage for output power calibration</td>
<td>0.218V</td>
<td>V</td>
<td>real</td>
<td>(0, ∞)</td>
</tr>
<tr>
<td>SamplingRate</td>
<td>sampling rate for output power calibration</td>
<td>(2048/224.0)MHz</td>
<td>Hz</td>
<td>real</td>
<td>(0, ∞)</td>
</tr>
</tbody>
</table>

**Pin Outputs**

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>PAL_signal</td>
<td>PAL signal</td>
<td>timed</td>
</tr>
</tbody>
</table>

**Notes/Equations**

1. The subnetwork model is used to generate PAL signal source. The PAL signal consists of color bars, FM sound (± 50kHz dev, 1kHz tone), and NICAM.

The schematic for this subnetwork is shown in Figure 3-44.
2. This PAL signal source is implemented according to European Air UHF band IV channel E21. E21 band limits are 470 to 478 MHz; center frequency is 474 MHz; video carrier frequency is 471.25 MHz; audio carrier frequency is 477.25 MHz; and, the NICAM carrier frequency is 477.802 MHz.

RF modulation occurs on the designated band.

\( F_{\text{Carrier}} \) is the center frequency of the designated PAL channel. Power sets the RF modulator output power. SamplingRate is the PAL signal bandwidth; its reciprocal is the RF modulation simulation time step.

References


DVB-T Components
Chapter 4: ISDB-T Components
ISDB-T Components

DTV_CDSCDecoder

Description  Complete differential set code (273,191) decoder
Library  DTV, ISDB-T
Class  SDFDTV_CDSCDecoder

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Default</th>
<th>Type</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>CodeLength</td>
<td>length of code bits</td>
<td>184</td>
<td>int</td>
<td>(0, 273)</td>
</tr>
<tr>
<td>Thresholds</td>
<td>thresholds for error detection</td>
<td>16 15 14 13 12 11 10 9</td>
<td>int array</td>
<td>(0, 17)</td>
</tr>
</tbody>
</table>

Pin Inputs

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>input</td>
<td>signal to be decoded</td>
<td>int</td>
</tr>
</tbody>
</table>

Pin Outputs

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>output</td>
<td>decoded signal</td>
<td>int</td>
</tr>
</tbody>
</table>

Notes/Equations

1. This model is used to perform complete differential set code (273,191) error correcting decoding over the input signal. After decoding, 184 or 273 bits are consumed at the input and 102 or 191 bits are produced at the output.

2. For the shortened code, the same number of 0 symbols is inserted into the same position as the CDSC coder; a CDSC decoder is used to decode the block. After decoding, the padded symbols are discarded, leaving the desired information symbols.

Generating Syndromes

4-2  DTV_CDSCDecoder
In the general situation of transmitting a codeword $C$ from a cyclic code by a uniform $q$-ary input, $q$-ary output channel. We seek the codeword that is nearest in Hamming distance to the received sequence, $R$. First, calculating the syndrome can be done with a simple feedback shift register similar to the encoder circuit. This is an improvement over building (or programming) a general-purpose multiplication of the received sequence $R$ with a parity check matrix $H$. Second, once the syndrome is calculated, it is possible to iteratively determine which code positions are to be corrected by cycling the syndrome register.

We represent the actions of the channel by

$$ R = C + E $$

where

$$ C = (c_0, c_1, ..., c_{n-1}) $$

represents the transmitted codeword and

$$ E = (e_0, e_1, ..., e_{n-1}) $$

is the error sequence

with both from $GF(q)$. In polynomial form, we express the channel action by

$$ r(x) = c(x) + e(x) $$

Now consider division of $r(x)$ by $g(x)$, generator polynomial for the code. This is motivated by recalling the valid codewords are exactly divisible by $g(x)$; that is, they produce zero remainder upon such division. We denote the remainder of this division as another polynomial, $s(x)$, the syndrome polynomial:

$$ s(x) = s_0 + s_1 x + ... + s_{n-k} x^{n-k} $$

$$ = r(x) \text{mod} g(x) = [c(x) + e(x)] \text{mod} g(x) $$

$$ = c(x) \text{mod} g(x) + e(x) \text{mod} g(x) $$

Eudid's division theorem implies that the syndrome polynomial is exactly determined by the error sequence, and $s(x)$ will have degree $n-k-1$ or less.

To calculate the syndrome polynomial, we require a circuit for calculating the remainder upon polynomial division. We provide a circuit and repeat the generic structure of the syndrome calculation in Figure 4-1, a device having $n-k$ $q$-ary register cells. The circuit is clocked $n$ times, at which time the syndrome vector (or polynomial) resides in the $(n-k)$-stage register, $s_0$ is the left-most memory cell.
Once the syndrome vector is determined, the syndrome vector is used to detect or correct errors.

**Using Syndromes to Decode Cyclic Codes**

---

**Figure 4-1. Syndrome-Forming Circuit**

CDSC (273,191) code is a one-order PG code. It can be decoded by one-step big numeric logic method. This decoding method is simpler than the method of decoding BCH codes. In this model, it only decodes the CDSC (273,191,d=18) and its shortened codes, such as (184,102).

Using the differential set of the CDSC (273,191), we can determine 17 orthogonal check sums. The differential set of CDSC (273,191,d=18) is \{0,18,24,46,50,67,103,112,115,126,128,159,166,167,186,196,201\}

The polynomial of the first orthogonal check sum is

\[
W_1(x) = x^{272} + x^{254} + x^{248} + x^{226} + x^{222} + x^{205} + x^{169} + x^{160} + x^{157} + x^{146} \\
+ x^{113} + x^{106} + x^{105} + x^{86} + x^{76} + x^{71}
\]

The polynomial of the second orthogonal check sum is

\[
W_2(x) = x^{18} W_1(x) \mod (x^{273} - 1) = x^{272} + x^{266} + x^{244} + x^{240} + x^{233} + x^{187} \\
+ x^{178} + x^{164} + x^{131} + x^{124} + x^{123} + x^{104} + x^{94} + x^{89} + x^{17}
\]

The polynomial of the third orthogonal check sum is

\[
W_3(x) = x^{24} W_1(x) \mod (x^{273} + 1) = x^{272} + x^{250} + x^{246} + x^{229} + x^{193} + x^{184} \\
+ x^{170} + x^{137} + x^{130} + x^{129} + x^{110} + x^{100} + x^{95} + x^3 + x^5 \ldots
\]
The polynomial of the third orthogonal check sum is 

\[ W_{17}(x) = x^{201}W_1(x) \mod (x^{273} + 1) = \ldots + \ldots + \ldots + \ldots + \ldots + x^{74} + x^{72} + x^{41} + x^{34} + x^{33} + x^{14} + x^4 \]

where the first part is omitted. According to the differential set, we get the 17 polynomial of the orthogonal check sums; then the 17 orthogonal check sums:

\[ A_1 = s_{76} \oplus s_{71} \]
\[ A_2 = s_{17} \]
\[ A_3 = s_{23} \oplus s_5 \]
\[ A_4 = s_{45} \oplus s_{27} \oplus s_{21} \]
\[ A_5 = s_{49} \oplus s_{31} \oplus s_{25} \oplus s_3 \]
\[ A_6 = s_{66} \oplus s_{42} \oplus s_{40} \oplus s_{16} \]
\[ A_7 = s_{78} \oplus s_{56} \oplus s_{52} \oplus s_{35} \]
\[ A_8 = s_{65} \oplus s_{61} \oplus s_{44} \oplus s_8 \]
\[ A_9 = s_{68} \oplus s_{64} \oplus s_{47} \oplus s_{11} \oplus s_2 \]
\[ A_{10} = s_{79} \oplus s_{75} \oplus s_{58} \oplus s_{22} \oplus s_{13} \oplus s_{10} \]
\[ A_{11} = s_{81} \oplus s_{77} \oplus s_{60} \oplus s_{24} \oplus s_{15} \oplus s_{12} \oplus s_1 \]
\[ A_{12} = s_{55} \oplus s_{46} \oplus s_{43} \oplus s_{32} \oplus s_{30} \]
\[ A_{13} = s_{62} \oplus s_{53} \oplus s_{50} \oplus s_{39} \oplus s_{37} \oplus s_6 \]
\[ A_{14} = s_{63} \oplus s_{54} \oplus s_{51} \oplus s_{40} \oplus s_{38} \oplus s_7 \oplus s_0 \]
\[ A_{15} = s_{73} \oplus s_{70} \oplus s_{59} \oplus s_{57} \oplus s_{26} \oplus s_{19} \oplus s_{18} \]
\[ A_{16} = s_{80} \oplus s_{69} \oplus s_{67} \oplus s_{36} \oplus s_{29} \oplus s_{28} \oplus s_9 \]
\[ A_{17} = s_{74} \oplus s_{72} \oplus s_{41} \oplus s_{34} \oplus s_{33} \oplus s_{14} \oplus s_4 \]

After the 17 orthogonal check sums are determined, the one-step numeric logic decoder decodes the received sequence based on the Thresholds value.

References

ISDB-T Components

DTV_CarrierRotator

Description  Particle rotation within segment
Library       DTV, ISDB-T
Class         SDFDTV_CarrierRotator

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Default</th>
<th>Type</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Carriers</td>
<td>block length of particles for one segment</td>
<td>384</td>
<td>int</td>
<td>(96, 192, 384)†</td>
</tr>
<tr>
<td>StartPoint</td>
<td>start particle number in segment</td>
<td>0 1 2 3</td>
<td>int array</td>
<td>(-∞, ∞)</td>
</tr>
<tr>
<td>Phase</td>
<td>initial phase of segment sequence</td>
<td>0</td>
<td>int</td>
<td>(-∞, ∞)</td>
</tr>
</tbody>
</table>

† Carriers = 96, 192, 384 for mode 1, 2, 3, respectively

Pin Inputs

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>input</td>
<td>symbols to be rotated</td>
<td>anytype</td>
</tr>
</tbody>
</table>

Pin Outputs

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>output</td>
<td>rotated symbols</td>
<td>anytype</td>
</tr>
</tbody>
</table>

Notes/Equations

1. This model is used to perform intra-segment carrier rotation for each segment in frequency interleaving.
2. This model implements the intra-segment rotation and permutation of the OFDM segment symbols shown in Figure 4-2.
ISDB-T Components

\[ S'_{(k + i \mod \text{Carriers})} = S_{i, 0, k} \]

\(0 \leq i \leq \text{Carriers} - 1\)

\(k\) is the segment number

\(S_{i, j, k}\) denotes the complex data of the \(k\)-th segment before inter-segment interleaving

\(S'_{i, j, k}\) denotes the complex data of the \(k\)-th segment after inter-segment interleaving

![Figure 4-2. Intra-Segment Carrier Rotation Interleaving](image)

(a) Intra-segment carrier rotation for Mode 1

(b) Intra-segment carrier rotation for Mode 2

(c) Intra-segment carrier rotation for Mode 3

References

**DTV_CarrierScrambler**

**Description**  
Carrier scrambler and descrambler

**Library**  
DTV, ISDB-T

**Class**  
SDFDTV_CarrierScrambler

### Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Default</th>
<th>Type</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Carriers</td>
<td>number of carriers for each segment in OFDM modulation mode</td>
<td>96</td>
<td>int</td>
<td>[96, 192, 384]†</td>
</tr>
<tr>
<td>Option</td>
<td>carrier option: Scramble, Descramble, Scramble, Descramble</td>
<td>Scramble</td>
<td>enum</td>
<td></td>
</tr>
<tr>
<td>Sequence</td>
<td>customized carrier mapping index, or empty to use the ISDB defaults</td>
<td>int array</td>
<td>[0, ∞)</td>
<td></td>
</tr>
</tbody>
</table>

† Carriers = 96, 192, 384 for mode 1, 2, 3, respectively

### Pin Inputs

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>input</td>
<td>symbols to be randomized</td>
<td>anytype</td>
</tr>
</tbody>
</table>

### Pin Outputs

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>output</td>
<td>randomized symbols</td>
<td>anytype</td>
</tr>
</tbody>
</table>

### Notes/Equations

1. This model is used to perform inter-segment carrier randomization in frequency interleaving for each segment.
2. This model assigns the OFDM symbols to each carrier within each segment in a pre-defined permutation order. Carrier permutation mapping from input to output according to modulation mode is described in Table 4-1 through Table 4-3. These mapping tables are used when Sequence is empty and Option indicates scrambling or de-scrambling. A customized mapping table can be used to implement a user-specified design, in which case Option is ignored.

### Table 4-1. Intra-Segment Carrier Randomization for Mode 1

<table>
<thead>
<tr>
<th>From</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
</tr>
</thead>
<tbody>
<tr>
<td>To</td>
<td>80</td>
<td>93</td>
<td>63</td>
<td>92</td>
<td>94</td>
<td>55</td>
<td>17</td>
<td>81</td>
<td>6</td>
<td>51</td>
<td>9</td>
<td>85</td>
</tr>
<tr>
<td>From</td>
<td>12</td>
<td>13</td>
<td>14</td>
<td>15</td>
<td>16</td>
<td>17</td>
<td>18</td>
<td>19</td>
<td>20</td>
<td>21</td>
<td>22</td>
<td>23</td>
</tr>
<tr>
<td>To</td>
<td>89</td>
<td>65</td>
<td>52</td>
<td>15</td>
<td>73</td>
<td>66</td>
<td>46</td>
<td>71</td>
<td>12</td>
<td>70</td>
<td>18</td>
<td>13</td>
</tr>
<tr>
<td>From</td>
<td>24</td>
<td>25</td>
<td>26</td>
<td>27</td>
<td>28</td>
<td>29</td>
<td>30</td>
<td>31</td>
<td>32</td>
<td>33</td>
<td>34</td>
<td>35</td>
</tr>
<tr>
<td>To</td>
<td>95</td>
<td>34</td>
<td>1</td>
<td>38</td>
<td>78</td>
<td>59</td>
<td>91</td>
<td>64</td>
<td>0</td>
<td>28</td>
<td>11</td>
<td>4</td>
</tr>
<tr>
<td>From</td>
<td>36</td>
<td>37</td>
<td>38</td>
<td>39</td>
<td>40</td>
<td>41</td>
<td>42</td>
<td>43</td>
<td>44</td>
<td>45</td>
<td>46</td>
<td>47</td>
</tr>
<tr>
<td>To</td>
<td>45</td>
<td>35</td>
<td>16</td>
<td>7</td>
<td>48</td>
<td>22</td>
<td>23</td>
<td>77</td>
<td>56</td>
<td>19</td>
<td>8</td>
<td>36</td>
</tr>
<tr>
<td>From</td>
<td>48</td>
<td>49</td>
<td>50</td>
<td>51</td>
<td>52</td>
<td>53</td>
<td>54</td>
<td>55</td>
<td>56</td>
<td>57</td>
<td>58</td>
<td>59</td>
</tr>
<tr>
<td>To</td>
<td>39</td>
<td>61</td>
<td>21</td>
<td>3</td>
<td>26</td>
<td>69</td>
<td>87</td>
<td>74</td>
<td>86</td>
<td>72</td>
<td>25</td>
<td></td>
</tr>
<tr>
<td>From</td>
<td>60</td>
<td>61</td>
<td>62</td>
<td>63</td>
<td>64</td>
<td>65</td>
<td>66</td>
<td>67</td>
<td>68</td>
<td>69</td>
<td>70</td>
<td>71</td>
</tr>
<tr>
<td>To</td>
<td>31</td>
<td>5</td>
<td>49</td>
<td>42</td>
<td>54</td>
<td>87</td>
<td>43</td>
<td>60</td>
<td>29</td>
<td>2</td>
<td>76</td>
<td>84</td>
</tr>
<tr>
<td>From</td>
<td>72</td>
<td>73</td>
<td>74</td>
<td>75</td>
<td>76</td>
<td>77</td>
<td>78</td>
<td>79</td>
<td>80</td>
<td>81</td>
<td>82</td>
<td>83</td>
</tr>
<tr>
<td>To</td>
<td>83</td>
<td>40</td>
<td>14</td>
<td>79</td>
<td>27</td>
<td>57</td>
<td>44</td>
<td>37</td>
<td>30</td>
<td>68</td>
<td>47</td>
<td>88</td>
</tr>
<tr>
<td>From</td>
<td>84</td>
<td>85</td>
<td>86</td>
<td>87</td>
<td>88</td>
<td>89</td>
<td>90</td>
<td>91</td>
<td>92</td>
<td>93</td>
<td>94</td>
<td>95</td>
</tr>
<tr>
<td>To</td>
<td>75</td>
<td>41</td>
<td>90</td>
<td>10</td>
<td>33</td>
<td>32</td>
<td>62</td>
<td>50</td>
<td>58</td>
<td>82</td>
<td>53</td>
<td>24</td>
</tr>
</tbody>
</table>

Values indicate carrier indexing; complex data indicated by the from carrier index is carried by the to carrier index.

### Table 4-2. Intra-Segment Carrier Randomization for Mode 2

<table>
<thead>
<tr>
<th>From</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
</tr>
</thead>
<tbody>
<tr>
<td>To</td>
<td>98</td>
<td>35</td>
<td>67</td>
<td>116</td>
<td>135</td>
<td>17</td>
<td>5</td>
<td>93</td>
<td>73</td>
<td>168</td>
<td>54</td>
<td>143</td>
</tr>
<tr>
<td>From</td>
<td>12</td>
<td>13</td>
<td>14</td>
<td>15</td>
<td>16</td>
<td>17</td>
<td>18</td>
<td>19</td>
<td>20</td>
<td>21</td>
<td>22</td>
<td>23</td>
</tr>
<tr>
<td>To</td>
<td>43</td>
<td>74</td>
<td>165</td>
<td>48</td>
<td>37</td>
<td>69</td>
<td>154</td>
<td>150</td>
<td>107</td>
<td>76</td>
<td>176</td>
<td>79</td>
</tr>
<tr>
<td>From</td>
<td>24</td>
<td>25</td>
<td>26</td>
<td>27</td>
<td>28</td>
<td>29</td>
<td>30</td>
<td>31</td>
<td>32</td>
<td>33</td>
<td>34</td>
<td>35</td>
</tr>
<tr>
<td>To</td>
<td>175</td>
<td>36</td>
<td>28</td>
<td>78</td>
<td>47</td>
<td>128</td>
<td>94</td>
<td>163</td>
<td>184</td>
<td>72</td>
<td>142</td>
<td>2</td>
</tr>
<tr>
<td>From</td>
<td>36</td>
<td>37</td>
<td>38</td>
<td>39</td>
<td>40</td>
<td>41</td>
<td>42</td>
<td>43</td>
<td>44</td>
<td>45</td>
<td>46</td>
<td>47</td>
</tr>
<tr>
<td>To</td>
<td>86</td>
<td>14</td>
<td>130</td>
<td>151</td>
<td>114</td>
<td>68</td>
<td>46</td>
<td>183</td>
<td>122</td>
<td>112</td>
<td>180</td>
<td>42</td>
</tr>
<tr>
<td>From</td>
<td>48</td>
<td>49</td>
<td>50</td>
<td>51</td>
<td>52</td>
<td>53</td>
<td>54</td>
<td>55</td>
<td>56</td>
<td>57</td>
<td>58</td>
<td>59</td>
</tr>
</tbody>
</table>
Table 4-2. Intra-Segment Carrier Randomization for Mode 2 (continued)

<table>
<thead>
<tr>
<th>To</th>
<th>105</th>
<th>97</th>
<th>33</th>
<th>134</th>
<th>177</th>
<th>84</th>
<th>170</th>
<th>45</th>
<th>187</th>
<th>38</th>
<th>167</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>From</td>
<td>60</td>
<td>61</td>
<td>62</td>
<td>63</td>
<td>64</td>
<td>65</td>
<td>66</td>
<td>67</td>
<td>68</td>
<td>69</td>
<td>70</td>
<td>71</td>
</tr>
<tr>
<td></td>
<td>189</td>
<td>51</td>
<td>117</td>
<td>156</td>
<td>161</td>
<td>25</td>
<td>89</td>
<td>125</td>
<td>139</td>
<td>24</td>
<td>19</td>
<td>57</td>
</tr>
<tr>
<td>From</td>
<td>72</td>
<td>73</td>
<td>74</td>
<td>75</td>
<td>76</td>
<td>77</td>
<td>78</td>
<td>79</td>
<td>80</td>
<td>81</td>
<td>82</td>
<td>83</td>
</tr>
<tr>
<td></td>
<td>71</td>
<td>39</td>
<td>77</td>
<td>191</td>
<td>88</td>
<td>85</td>
<td>0</td>
<td>162</td>
<td>181</td>
<td>113</td>
<td>140</td>
<td>61</td>
</tr>
<tr>
<td>From</td>
<td>84</td>
<td>85</td>
<td>86</td>
<td>87</td>
<td>88</td>
<td>89</td>
<td>90</td>
<td>91</td>
<td>92</td>
<td>93</td>
<td>94</td>
<td>95</td>
</tr>
<tr>
<td></td>
<td>75</td>
<td>82</td>
<td>101</td>
<td>174</td>
<td>118</td>
<td>20</td>
<td>136</td>
<td>3</td>
<td>121</td>
<td>190</td>
<td>120</td>
<td>92</td>
</tr>
<tr>
<td>From</td>
<td>96</td>
<td>97</td>
<td>98</td>
<td>99</td>
<td>100</td>
<td>101</td>
<td>102</td>
<td>103</td>
<td>104</td>
<td>105</td>
<td>106</td>
<td>107</td>
</tr>
<tr>
<td></td>
<td>160</td>
<td>52</td>
<td>153</td>
<td>127</td>
<td>65</td>
<td>60</td>
<td>133</td>
<td>147</td>
<td>131</td>
<td>87</td>
<td>22</td>
<td>58</td>
</tr>
<tr>
<td>From</td>
<td>108</td>
<td>109</td>
<td>110</td>
<td>111</td>
<td>112</td>
<td>113</td>
<td>114</td>
<td>115</td>
<td>116</td>
<td>117</td>
<td>118</td>
<td>119</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>111</td>
<td>141</td>
<td>83</td>
<td>49</td>
<td>132</td>
<td>12</td>
<td>155</td>
<td>146</td>
<td>102</td>
<td>164</td>
<td>66</td>
</tr>
<tr>
<td>From</td>
<td>120</td>
<td>121</td>
<td>122</td>
<td>123</td>
<td>124</td>
<td>125</td>
<td>126</td>
<td>127</td>
<td>128</td>
<td>129</td>
<td>130</td>
<td>131</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>62</td>
<td>178</td>
<td>15</td>
<td>182</td>
<td>98</td>
<td>80</td>
<td>119</td>
<td>23</td>
<td>6</td>
<td>166</td>
<td>56</td>
</tr>
<tr>
<td>From</td>
<td>132</td>
<td>133</td>
<td>134</td>
<td>135</td>
<td>136</td>
<td>137</td>
<td>138</td>
<td>139</td>
<td>140</td>
<td>141</td>
<td>142</td>
<td>143</td>
</tr>
<tr>
<td></td>
<td>99</td>
<td>123</td>
<td>138</td>
<td>137</td>
<td>21</td>
<td>145</td>
<td>185</td>
<td>18</td>
<td>70</td>
<td>129</td>
<td>95</td>
<td>90</td>
</tr>
<tr>
<td>From</td>
<td>144</td>
<td>145</td>
<td>146</td>
<td>147</td>
<td>148</td>
<td>149</td>
<td>150</td>
<td>151</td>
<td>152</td>
<td>153</td>
<td>154</td>
<td>155</td>
</tr>
<tr>
<td></td>
<td>149</td>
<td>109</td>
<td>124</td>
<td>50</td>
<td>11</td>
<td>152</td>
<td>4</td>
<td>31</td>
<td>172</td>
<td>40</td>
<td>13</td>
<td>32</td>
</tr>
<tr>
<td>From</td>
<td>156</td>
<td>157</td>
<td>158</td>
<td>159</td>
<td>160</td>
<td>161</td>
<td>162</td>
<td>163</td>
<td>164</td>
<td>165</td>
<td>166</td>
<td>167</td>
</tr>
<tr>
<td></td>
<td>55</td>
<td>159</td>
<td>41</td>
<td>8</td>
<td>7</td>
<td>144</td>
<td>16</td>
<td>26</td>
<td>173</td>
<td>81</td>
<td>44</td>
<td>103</td>
</tr>
<tr>
<td>From</td>
<td>168</td>
<td>169</td>
<td>170</td>
<td>171</td>
<td>172</td>
<td>173</td>
<td>174</td>
<td>175</td>
<td>176</td>
<td>177</td>
<td>178</td>
<td>179</td>
</tr>
<tr>
<td></td>
<td>64</td>
<td>9</td>
<td>30</td>
<td>157</td>
<td>126</td>
<td>179</td>
<td>148</td>
<td>63</td>
<td>188</td>
<td>171</td>
<td>106</td>
<td>104</td>
</tr>
<tr>
<td>From</td>
<td>180</td>
<td>181</td>
<td>182</td>
<td>183</td>
<td>184</td>
<td>185</td>
<td>186</td>
<td>187</td>
<td>188</td>
<td>189</td>
<td>190</td>
<td>191</td>
</tr>
<tr>
<td></td>
<td>158</td>
<td>115</td>
<td>34</td>
<td>186</td>
<td>29</td>
<td>108</td>
<td>53</td>
<td>91</td>
<td>169</td>
<td>110</td>
<td>27</td>
<td>59</td>
</tr>
</tbody>
</table>

Values indicate carrier indexing; complex data indicated by the From carrier index is carried by the To carrier index.

Table 4-3. Intra-Segment Carrier Randomization for Mode 3

<table>
<thead>
<tr>
<th>From</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
</tr>
</thead>
<tbody>
<tr>
<td>To</td>
<td>62</td>
<td>13</td>
<td>11</td>
<td>285</td>
<td>336</td>
<td>365</td>
<td>220</td>
<td>226</td>
<td>92</td>
<td>56</td>
<td>46</td>
<td></td>
</tr>
<tr>
<td>From</td>
<td>12</td>
<td>13</td>
<td>14</td>
<td>15</td>
<td>16</td>
<td>17</td>
<td>18</td>
<td>19</td>
<td>20</td>
<td>21</td>
<td>22</td>
<td>23</td>
</tr>
<tr>
<td>To</td>
<td>120</td>
<td>175</td>
<td>298</td>
<td>352</td>
<td>172</td>
<td>235</td>
<td>53</td>
<td>164</td>
<td>368</td>
<td>187</td>
<td>125</td>
<td>82</td>
</tr>
<tr>
<td>From</td>
<td>24</td>
<td>25</td>
<td>26</td>
<td>27</td>
<td>28</td>
<td>29</td>
<td>30</td>
<td>31</td>
<td>32</td>
<td>33</td>
<td>34</td>
<td>35</td>
</tr>
<tr>
<td>To</td>
<td>5</td>
<td>45</td>
<td>173</td>
<td>258</td>
<td>135</td>
<td>182</td>
<td>141</td>
<td>273</td>
<td>126</td>
<td>264</td>
<td>286</td>
<td>88</td>
</tr>
<tr>
<td>From</td>
<td>36</td>
<td>37</td>
<td>38</td>
<td>39</td>
<td>40</td>
<td>41</td>
<td>42</td>
<td>43</td>
<td>44</td>
<td>45</td>
<td>46</td>
<td>47</td>
</tr>
<tr>
<td>To</td>
<td>233</td>
<td>61</td>
<td>249</td>
<td>367</td>
<td>310</td>
<td>179</td>
<td>155</td>
<td>57</td>
<td>123</td>
<td>208</td>
<td>14</td>
<td>227</td>
</tr>
<tr>
<td>From</td>
<td>48</td>
<td>49</td>
<td>50</td>
<td>51</td>
<td>52</td>
<td>53</td>
<td>54</td>
<td>55</td>
<td>56</td>
<td>57</td>
<td>58</td>
<td>59</td>
</tr>
</tbody>
</table>
Table 4-3. Intra-Segment Carrier Randomization for Mode 3 (continued)

<table>
<thead>
<tr>
<th>From</th>
<th>100</th>
<th>311</th>
<th>205</th>
<th>79</th>
<th>184</th>
<th>185</th>
<th>328</th>
<th>77</th>
<th>115</th>
<th>277</th>
<th>112</th>
<th>20</th>
</tr>
</thead>
<tbody>
<tr>
<td>To</td>
<td>60</td>
<td>61</td>
<td>62</td>
<td>63</td>
<td>64</td>
<td>65</td>
<td>66</td>
<td>67</td>
<td>68</td>
<td>69</td>
<td>70</td>
<td>71</td>
</tr>
<tr>
<td>From</td>
<td>199</td>
<td>178</td>
<td>143</td>
<td>152</td>
<td>215</td>
<td>204</td>
<td>139</td>
<td>234</td>
<td>358</td>
<td>192</td>
<td>309</td>
<td>183</td>
</tr>
<tr>
<td>To</td>
<td>72</td>
<td>73</td>
<td>74</td>
<td>75</td>
<td>76</td>
<td>77</td>
<td>78</td>
<td>79</td>
<td>80</td>
<td>81</td>
<td>82</td>
<td>83</td>
</tr>
<tr>
<td>From</td>
<td>81</td>
<td>129</td>
<td>256</td>
<td>314</td>
<td>101</td>
<td>43</td>
<td>97</td>
<td>324</td>
<td>142</td>
<td>157</td>
<td>90</td>
<td>214</td>
</tr>
<tr>
<td>To</td>
<td>84</td>
<td>85</td>
<td>86</td>
<td>87</td>
<td>88</td>
<td>89</td>
<td>90</td>
<td>91</td>
<td>92</td>
<td>93</td>
<td>94</td>
<td>95</td>
</tr>
<tr>
<td>From</td>
<td>102</td>
<td>29</td>
<td>303</td>
<td>363</td>
<td>261</td>
<td>31</td>
<td>22</td>
<td>52</td>
<td>305</td>
<td>301</td>
<td>293</td>
<td>177</td>
</tr>
<tr>
<td>To</td>
<td>96</td>
<td>97</td>
<td>98</td>
<td>99</td>
<td>100</td>
<td>101</td>
<td>102</td>
<td>103</td>
<td>104</td>
<td>105</td>
<td>106</td>
<td>107</td>
</tr>
<tr>
<td>From</td>
<td>116</td>
<td>296</td>
<td>85</td>
<td>196</td>
<td>191</td>
<td>114</td>
<td>58</td>
<td>198</td>
<td>16</td>
<td>167</td>
<td>145</td>
<td>119</td>
</tr>
<tr>
<td>To</td>
<td>108</td>
<td>109</td>
<td>110</td>
<td>111</td>
<td>112</td>
<td>113</td>
<td>114</td>
<td>115</td>
<td>116</td>
<td>117</td>
<td>118</td>
<td>119</td>
</tr>
<tr>
<td>From</td>
<td>245</td>
<td>113</td>
<td>295</td>
<td>193</td>
<td>232</td>
<td>17</td>
<td>108</td>
<td>283</td>
<td>246</td>
<td>64</td>
<td>237</td>
<td>189</td>
</tr>
<tr>
<td>To</td>
<td>120</td>
<td>121</td>
<td>122</td>
<td>123</td>
<td>124</td>
<td>125</td>
<td>126</td>
<td>127</td>
<td>128</td>
<td>129</td>
<td>130</td>
<td>131</td>
</tr>
<tr>
<td>From</td>
<td>128</td>
<td>373</td>
<td>302</td>
<td>320</td>
<td>239</td>
<td>335</td>
<td>356</td>
<td>39</td>
<td>347</td>
<td>351</td>
<td>73</td>
<td>158</td>
</tr>
<tr>
<td>To</td>
<td>132</td>
<td>133</td>
<td>134</td>
<td>135</td>
<td>136</td>
<td>137</td>
<td>138</td>
<td>139</td>
<td>140</td>
<td>141</td>
<td>142</td>
<td>143</td>
</tr>
<tr>
<td>From</td>
<td>276</td>
<td>243</td>
<td>99</td>
<td>38</td>
<td>287</td>
<td>3</td>
<td>330</td>
<td>153</td>
<td>315</td>
<td>117</td>
<td>289</td>
<td>213</td>
</tr>
<tr>
<td>To</td>
<td>144</td>
<td>145</td>
<td>146</td>
<td>147</td>
<td>148</td>
<td>149</td>
<td>150</td>
<td>151</td>
<td>152</td>
<td>153</td>
<td>154</td>
<td>155</td>
</tr>
<tr>
<td>From</td>
<td>210</td>
<td>149</td>
<td>383</td>
<td>337</td>
<td>339</td>
<td>151</td>
<td>241</td>
<td>321</td>
<td>217</td>
<td>30</td>
<td>334</td>
<td>161</td>
</tr>
<tr>
<td>To</td>
<td>156</td>
<td>157</td>
<td>158</td>
<td>159</td>
<td>160</td>
<td>161</td>
<td>162</td>
<td>163</td>
<td>164</td>
<td>165</td>
<td>166</td>
<td>167</td>
</tr>
<tr>
<td>From</td>
<td>322</td>
<td>49</td>
<td>176</td>
<td>359</td>
<td>12</td>
<td>346</td>
<td>60</td>
<td>28</td>
<td>229</td>
<td>265</td>
<td>288</td>
<td>225</td>
</tr>
<tr>
<td>To</td>
<td>168</td>
<td>169</td>
<td>170</td>
<td>171</td>
<td>172</td>
<td>173</td>
<td>174</td>
<td>175</td>
<td>176</td>
<td>177</td>
<td>178</td>
<td>179</td>
</tr>
<tr>
<td>From</td>
<td>382</td>
<td>59</td>
<td>181</td>
<td>182</td>
<td>183</td>
<td>184</td>
<td>185</td>
<td>186</td>
<td>187</td>
<td>188</td>
<td>189</td>
<td>190</td>
</tr>
<tr>
<td>To</td>
<td>180</td>
<td>181</td>
<td>182</td>
<td>183</td>
<td>184</td>
<td>185</td>
<td>186</td>
<td>187</td>
<td>188</td>
<td>189</td>
<td>190</td>
<td>191</td>
</tr>
<tr>
<td>From</td>
<td>44</td>
<td>369</td>
<td>268</td>
<td>257</td>
<td>323</td>
<td>55</td>
<td>317</td>
<td>381</td>
<td>121</td>
<td>360</td>
<td>260</td>
<td>275</td>
</tr>
<tr>
<td>To</td>
<td>192</td>
<td>193</td>
<td>194</td>
<td>195</td>
<td>196</td>
<td>197</td>
<td>198</td>
<td>199</td>
<td>200</td>
<td>201</td>
<td>202</td>
<td>203</td>
</tr>
<tr>
<td>From</td>
<td>190</td>
<td>19</td>
<td>63</td>
<td>18</td>
<td>248</td>
<td>9</td>
<td>240</td>
<td>211</td>
<td>150</td>
<td>230</td>
<td>332</td>
<td>231</td>
</tr>
<tr>
<td>To</td>
<td>204</td>
<td>205</td>
<td>206</td>
<td>207</td>
<td>208</td>
<td>209</td>
<td>210</td>
<td>211</td>
<td>212</td>
<td>213</td>
<td>214</td>
<td>215</td>
</tr>
<tr>
<td>From</td>
<td>71</td>
<td>255</td>
<td>350</td>
<td>355</td>
<td>83</td>
<td>87</td>
<td>154</td>
<td>218</td>
<td>138</td>
<td>269</td>
<td>348</td>
<td>130</td>
</tr>
<tr>
<td>To</td>
<td>216</td>
<td>217</td>
<td>218</td>
<td>219</td>
<td>220</td>
<td>221</td>
<td>222</td>
<td>223</td>
<td>224</td>
<td>225</td>
<td>226</td>
<td>227</td>
</tr>
<tr>
<td>From</td>
<td>160</td>
<td>278</td>
<td>377</td>
<td>216</td>
<td>236</td>
<td>308</td>
<td>223</td>
<td>254</td>
<td>25</td>
<td>98</td>
<td>300</td>
<td>201</td>
</tr>
<tr>
<td>To</td>
<td>228</td>
<td>229</td>
<td>230</td>
<td>231</td>
<td>232</td>
<td>233</td>
<td>234</td>
<td>235</td>
<td>236</td>
<td>237</td>
<td>238</td>
<td>239</td>
</tr>
<tr>
<td>From</td>
<td>137</td>
<td>219</td>
<td>36</td>
<td>325</td>
<td>124</td>
<td>66</td>
<td>353</td>
<td>169</td>
<td>21</td>
<td>35</td>
<td>107</td>
<td>50</td>
</tr>
<tr>
<td>To</td>
<td>240</td>
<td>241</td>
<td>242</td>
<td>243</td>
<td>244</td>
<td>245</td>
<td>246</td>
<td>247</td>
<td>248</td>
<td>249</td>
<td>250</td>
<td>251</td>
</tr>
<tr>
<td>From</td>
<td>106</td>
<td>333</td>
<td>326</td>
<td>262</td>
<td>252</td>
<td>271</td>
<td>263</td>
<td>372</td>
<td>136</td>
<td>0</td>
<td>356</td>
<td>206</td>
</tr>
<tr>
<td>To</td>
<td>252</td>
<td>253</td>
<td>254</td>
<td>255</td>
<td>256</td>
<td>257</td>
<td>258</td>
<td>259</td>
<td>260</td>
<td>261</td>
<td>262</td>
<td>263</td>
</tr>
<tr>
<td>From</td>
<td>159</td>
<td>122</td>
<td>188</td>
<td>6</td>
<td>284</td>
<td>96</td>
<td>26</td>
<td>200</td>
<td>197</td>
<td>186</td>
<td>345</td>
<td>340</td>
</tr>
<tr>
<td>To</td>
<td>264</td>
<td>265</td>
<td>266</td>
<td>267</td>
<td>268</td>
<td>269</td>
<td>270</td>
<td>271</td>
<td>272</td>
<td>273</td>
<td>274</td>
<td>275</td>
</tr>
</tbody>
</table>
Table 4-3. Intra-Segment Carrier Randomization for Mode 3 (continued)

<table>
<thead>
<tr>
<th>To</th>
<th>349</th>
<th>103</th>
<th>84</th>
<th>228</th>
<th>212</th>
<th>2</th>
<th>67</th>
<th>318</th>
<th>1</th>
<th>74</th>
<th>342</th>
<th>166</th>
</tr>
</thead>
<tbody>
<tr>
<td>From</td>
<td>276</td>
<td>277</td>
<td>278</td>
<td>279</td>
<td>280</td>
<td>281</td>
<td>282</td>
<td>283</td>
<td>284</td>
<td>285</td>
<td>286</td>
<td>287</td>
</tr>
<tr>
<td>To</td>
<td>194</td>
<td>33</td>
<td>68</td>
<td>267</td>
<td>111</td>
<td>118</td>
<td>140</td>
<td>195</td>
<td>105</td>
<td>202</td>
<td>291</td>
<td>259</td>
</tr>
<tr>
<td>From</td>
<td>288</td>
<td>289</td>
<td>290</td>
<td>291</td>
<td>292</td>
<td>293</td>
<td>294</td>
<td>295</td>
<td>296</td>
<td>297</td>
<td>298</td>
<td>299</td>
</tr>
<tr>
<td>To</td>
<td>23</td>
<td>171</td>
<td>65</td>
<td>281</td>
<td>24</td>
<td>165</td>
<td>8</td>
<td>94</td>
<td>222</td>
<td>331</td>
<td>34</td>
<td>238</td>
</tr>
<tr>
<td>From</td>
<td>300</td>
<td>301</td>
<td>302</td>
<td>303</td>
<td>304</td>
<td>305</td>
<td>306</td>
<td>307</td>
<td>308</td>
<td>309</td>
<td>310</td>
<td>311</td>
</tr>
<tr>
<td>To</td>
<td>364</td>
<td>376</td>
<td>266</td>
<td>89</td>
<td>80</td>
<td>253</td>
<td>163</td>
<td>280</td>
<td>247</td>
<td>4</td>
<td>362</td>
<td>379</td>
</tr>
<tr>
<td>From</td>
<td>312</td>
<td>313</td>
<td>314</td>
<td>315</td>
<td>316</td>
<td>317</td>
<td>318</td>
<td>319</td>
<td>320</td>
<td>321</td>
<td>322</td>
<td>323</td>
</tr>
<tr>
<td>To</td>
<td>290</td>
<td>279</td>
<td>54</td>
<td>78</td>
<td>180</td>
<td>72</td>
<td>316</td>
<td>282</td>
<td>131</td>
<td>207</td>
<td>343</td>
<td>370</td>
</tr>
<tr>
<td>From</td>
<td>324</td>
<td>325</td>
<td>326</td>
<td>327</td>
<td>328</td>
<td>329</td>
<td>330</td>
<td>331</td>
<td>332</td>
<td>333</td>
<td>334</td>
<td>335</td>
</tr>
<tr>
<td>To</td>
<td>306</td>
<td>221</td>
<td>132</td>
<td>7</td>
<td>148</td>
<td>299</td>
<td>168</td>
<td>224</td>
<td>48</td>
<td>47</td>
<td>357</td>
<td>313</td>
</tr>
<tr>
<td>From</td>
<td>336</td>
<td>337</td>
<td>338</td>
<td>339</td>
<td>340</td>
<td>341</td>
<td>342</td>
<td>343</td>
<td>344</td>
<td>345</td>
<td>346</td>
<td>347</td>
</tr>
<tr>
<td>To</td>
<td>75</td>
<td>104</td>
<td>70</td>
<td>147</td>
<td>40</td>
<td>110</td>
<td>374</td>
<td>69</td>
<td>146</td>
<td>37</td>
<td>375</td>
<td>354</td>
</tr>
<tr>
<td>From</td>
<td>348</td>
<td>349</td>
<td>350</td>
<td>351</td>
<td>352</td>
<td>353</td>
<td>354</td>
<td>355</td>
<td>356</td>
<td>357</td>
<td>358</td>
<td>359</td>
</tr>
<tr>
<td>To</td>
<td>174</td>
<td>41</td>
<td>32</td>
<td>304</td>
<td>307</td>
<td>312</td>
<td>15</td>
<td>272</td>
<td>134</td>
<td>242</td>
<td>203</td>
<td>209</td>
</tr>
<tr>
<td>From</td>
<td>360</td>
<td>361</td>
<td>362</td>
<td>363</td>
<td>364</td>
<td>365</td>
<td>366</td>
<td>367</td>
<td>368</td>
<td>369</td>
<td>370</td>
<td>371</td>
</tr>
<tr>
<td>To</td>
<td>380</td>
<td>162</td>
<td>297</td>
<td>327</td>
<td>10</td>
<td>93</td>
<td>42</td>
<td>259</td>
<td>156</td>
<td>338</td>
<td>292</td>
<td>144</td>
</tr>
<tr>
<td>From</td>
<td>372</td>
<td>373</td>
<td>374</td>
<td>375</td>
<td>376</td>
<td>377</td>
<td>378</td>
<td>379</td>
<td>380</td>
<td>381</td>
<td>382</td>
<td>383</td>
</tr>
<tr>
<td>To</td>
<td>378</td>
<td>294</td>
<td>329</td>
<td>127</td>
<td>270</td>
<td>76</td>
<td>95</td>
<td>91</td>
<td>244</td>
<td>274</td>
<td>27</td>
<td>51</td>
</tr>
</tbody>
</table>

Values indicate carrier indexed, and complex data indicated by the carrier index. From is carried by the carrier index To.

References

ISDB-T Components

**DTV_ChEstimator**

**Description**  OFDM symbol channel estimator and channel interpolator for ISDB-T

**Library**  DTV, ISDB-T

**Class**  SDFDTV_ChEstimator

**Parameters**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Default</th>
<th>Type</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Layers</td>
<td>number of layers</td>
<td>1</td>
<td>int</td>
<td>1 for 1-layer; 2 for 2-layer; 3 for 3-layer systems</td>
</tr>
<tr>
<td>Carriers</td>
<td>number of carriers in one OFDM symbol</td>
<td>432</td>
<td>int</td>
<td>108 for mode 1; 216 for mode 2; 432 for mode 3</td>
</tr>
<tr>
<td>Segments</td>
<td>number of segments in each layer</td>
<td>13</td>
<td>int array</td>
<td>[1, 13]</td>
</tr>
<tr>
<td>InSequence</td>
<td>segment sequence at input</td>
<td>11 9 7 5 3 1 0 2 4 6 8 10 12</td>
<td>int array</td>
<td>&quot;0, 1,...,Nseg-1&quot;</td>
</tr>
<tr>
<td>OutSequence</td>
<td>segment sequence at output</td>
<td>0 1 2 3 4 5 6 7 8 9 10 11 12</td>
<td>int array</td>
<td>&quot;..., 1,0,2,...&quot;</td>
</tr>
<tr>
<td>Order</td>
<td>FFT points=2^Order</td>
<td>13</td>
<td>int</td>
<td>[1, ∞)</td>
</tr>
<tr>
<td>CPnumber</td>
<td>number of continual pilots in each segment for each layer</td>
<td>0</td>
<td>int array</td>
<td>{1, 0}</td>
</tr>
<tr>
<td>SPnumber</td>
<td>number of scattered pilots in each segment</td>
<td>36</td>
<td>int array</td>
<td>[1, ∞)</td>
</tr>
<tr>
<td>SPperiod</td>
<td>distance in carriers between nearby scattered pilots in each segment for each layer</td>
<td>12</td>
<td>int array</td>
<td>[0, ∞)</td>
</tr>
<tr>
<td>SPstart</td>
<td>start position of scattered pilots in carriers</td>
<td>0</td>
<td>int array</td>
<td>[1, ∞)</td>
</tr>
<tr>
<td>SPoffset</td>
<td>offset value of SPstart in each symbol</td>
<td>3</td>
<td>int array</td>
<td>[0, ∞)</td>
</tr>
<tr>
<td>SPphase</td>
<td>initial phase of scattered pilots for each layer</td>
<td>0</td>
<td>int array</td>
<td>[0, SPperiod/SPoffset[i-1]]</td>
</tr>
</tbody>
</table>

4-14  DTV_ChEstimator
Notes/Equations

1. The model is used to adjust the transmission spectrum, linear channel estimation, channel interpolation based on the pilots channel. It outputs the segments data and corresponding subchannel estimation according to the OutSequence.

2. Segments is the array of number of segments in each layer. Segments ≥1, but the sum of each component in Segments ≤13. For example: Segments= "1,5,6" is correct; Segments= "0,7" or Segments= "1,7,8" are not correct.

3. InSequence is [0, Nseg-1], where Nseg=Segments[0]+...+Segments[Layer-1] is the sum of each component in Segments. According to ISDB-T, the InSequence is "0,1,2,...,Nseg-1" after Segments is determined.

4. OutSequence is [0, Nseg-1]. According to ISDB-T, the OutSequence is "... ,1, 0, 2, ..." after Segments is determined. For example: if Segments= "1,4", OutSequence is "3,1,0,2,4"; if Segments= "1,4,5", OutSequence is "9,7,5,3,1,0,2,4,6,8".

5. Order is the order of FFT. It must satisfy $2^{Order} \geq $ Carriers × Nseg.

6. CPnumber is 0 or 1 according to ISDB-T. If the modulation mode is DQPSK in one layer, its corresponding component in CPnumber is 1 per one segment, otherwise, its corresponding component in CPnumber is 0. For example: in a 2-layer system, the first layer is DQPSK modulation, the second layer is 16-QAM modulation, the CPnumber= "1, 0".

7. SPnumber = 0, 9, 18 or 36 according to ISDB-T. If the modulation mode is:
   • differential modulation in one layer, its corresponding component in SPnumber is 0.
ISDB-T Components

- coherent modulation in one layer, its corresponding component in SPnumber is 9, 18 or 36 of mode 1, mode 2, or mode 3 per one segment, respectively.

For example, in a 3-layer mode 3 system, the first and third layers are coherent modulation and the second layer is differential modulation; SPnumber = "36, 0, 36".

8. SPperiod = 0 or 12 according to ISDB-T. If the modulation mode is:
   - differential modulation in one layer, its corresponding component in SPperiod is 0
   - coherent modulation in one layer, its corresponding component in SPperiod is 12.

   For example, in a 3-layer mode 3 system, the first and third layers are coherent modulation and the second layer is differential modulation; SPperiod = "12, 0, 12".

9. SPstart = 0 according to ISDB-T.

10. SPoffset = 0 or 3 according to ISDB-T. If the modulation mode is:
    - differential modulation in one layer, its corresponding component in SPoffset is 0
    - coherent modulation in one layer, its corresponding component in SPoffset is 3

    For example, in a 3-layer mode 3 system, the first and third layers are coherent modulation and the second layer is differential modulation; SPoffset = "3, 0, 3".

11. SPphase = 3 in the OFDM receiver if the modulation mode of the corresponding layer is coherent modulation (because DTV_MLEstimator makes one OFDM symbol delay); otherwise, SPphase = 0.

    For example, in a 3-layer mode 3 system, the first and third layers are coherent modulation and the second layer is differential modulation; SPphase = "3, 0, 3".

12. Using IntState of Carriers, CPnumber and SPnumber, this model determines the number of CP (continual pilot) and SP (scattered pilots) in each segment in every layer. According to ISDB-T, if the modulation mode in one layer is:
    - differential modulation, there is one CP and no SP in every segment.
• coherent modulation, there are 9, 18, 36 SPs in each segment in one layer, for mode 1, mode 2, and mode 3, respectively, and no any CP in any segment.

PRBS sequences in each segment are generated according to Figure 4-3 and the initial sets of PRBS register in Table 4-4. The PRBS is initialized so that the first output bit from the PRBS coincides with the first active carrier. A new value is generated by the PRBS on every carrier used in each segment (whether or not it is a pilot).

The position of one CP in the differential modulation segment is always zero.

The positions of the scattered pilots in the coherent modulation segments are generated as follows. For the symbol of index $l$ (ranging from 0 to 203), carriers for which index $k$ belongs to subset $\{k \mod 4 + 12 p \mid p \text{ integer}, p \geq 0, k \in [0, \text{Length} \times \text{Segments}[i])\}$ are SP positions, where $i$ is 0, 1, 2, corresponding to the coherent modulation segments layers. Five parameters control the SP positions.

• SPnumber determines the number of scattered pilots in each segment: 9 in mode 1 (Length=108), 18 in mode 2 (Length=216), and 36 in mode 3 (Length=432).

• SPperiod =12, SPstart = 0, SPOffset = 3 and SPPhase=0 in all modes according to ISDB-T.

After determining all CP and SP positions in each corresponding segment in every layer, we get the pilots value from the PRBS sequence. So, we can get the channel estimation in this CP and SP pilots.

$$h(i) = \frac{x(i)}{\text{PilotValue}(i)}$$

where $h(i)$ is the channel estimation $x(i)$ is the received signal from channel after FFT and $\text{PilotValue}(i)$ is the PRBS value corresponding to CP and SP position in each segment.

After getting the subchannel estimation in the CP and SP pilots position, we use the linear interpolation algorithm to get all active subchannel estimation:

$$h(k) = \alpha h(i) + (1 - \alpha) h(j)$$

$$\alpha = \frac{j - k}{j - i}$$
ISDB-T Components

where \( i \leq h \leq j \), and \( h(i) \) and \( h(j) \) are the subchannel estimation of the CP or SP pilots.

The transmission spectrum adjustment is made in the DTV_LoadIFFTBuff. In the receiver, the transmission spectrum must be inversed. The InSequence and OutSequence must be identical to DTV_LoadIFFTBuff. The layer segment data is output by the order of OutSequence. The inserted zeros in the DTV_LoadIFFTBuff are discarded. The active carriers and its corresponding subchannel carriers are output for use in the equalizer model.

Figure 4-3. Generation of PRBS Sequence

Table 4-4. Initial Sets of PRBS Register

<table>
<thead>
<tr>
<th>Segment Number</th>
<th>Initial Sets for Mode 1†</th>
<th>Initial Sets for Mode 2†</th>
<th>Initial Sets for Mode 3†</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>111111111111</td>
<td>111111111111</td>
<td>111111111111</td>
</tr>
<tr>
<td>9</td>
<td>110110011111</td>
<td>011010111110</td>
<td>11011100101</td>
</tr>
<tr>
<td>7</td>
<td>011010111110</td>
<td>11011100101</td>
<td>10010100000</td>
</tr>
<tr>
<td>5</td>
<td>010001011110</td>
<td>11001000100</td>
<td>01110000100</td>
</tr>
<tr>
<td>3</td>
<td>11011100101</td>
<td>10010100000</td>
<td>01000110100</td>
</tr>
<tr>
<td>1</td>
<td>001001111010</td>
<td>00001011000</td>
<td>11100110110</td>
</tr>
<tr>
<td>0</td>
<td>11001000100</td>
<td>01110000101</td>
<td>00100001011</td>
</tr>
<tr>
<td>2</td>
<td>00010000100</td>
<td>00001001000</td>
<td>11100111101</td>
</tr>
<tr>
<td>4</td>
<td>10010100000</td>
<td>00100011001</td>
<td>01101010011</td>
</tr>
<tr>
<td>6</td>
<td>11110110000</td>
<td>01110111001</td>
<td>10111100101</td>
</tr>
<tr>
<td>8</td>
<td>00001011000</td>
<td>11100110110</td>
<td>01100010010</td>
</tr>
<tr>
<td>10</td>
<td>10100100111</td>
<td>00101010001</td>
<td>1110100101</td>
</tr>
<tr>
<td>12</td>
<td>01110001001</td>
<td>0010001011</td>
<td>0010011100</td>
</tr>
</tbody>
</table>

†Degree from 0 to 10 in Figure 4-3.

References
ISDB-T Components

**DTV_DemuxCohSegs**

**Description**  OFDM de-segment for coherent modulation

**Library**  DTV, ISDB-T

**Class**  SDFDTV_DemuxCohSegs

### Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Default</th>
<th>Type</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Carriers</td>
<td>number of carriers in one segment</td>
<td>432</td>
<td>int</td>
<td>108 for mode 1; 216 for mode 2; 432 for mode 3</td>
</tr>
<tr>
<td>Segments</td>
<td>number of segments</td>
<td>1</td>
<td>int</td>
<td>[1, 13]</td>
</tr>
<tr>
<td>Start_Seg</td>
<td>initial number of segment (0 to Segments-1)</td>
<td>0</td>
<td>int</td>
<td>[0, 12]</td>
</tr>
<tr>
<td>SPNumber</td>
<td>number of scattered pilots in each segment</td>
<td>36</td>
<td>int</td>
<td>[0, ∞]†††</td>
</tr>
<tr>
<td>SPPeriod</td>
<td>distance in carriers between nearby scattered pilots</td>
<td>12</td>
<td>int</td>
<td>(0, ∞)†††</td>
</tr>
<tr>
<td>SPStart</td>
<td>start position of scattered pilots in carriers</td>
<td>0</td>
<td>int</td>
<td>[0, ∞)††</td>
</tr>
<tr>
<td>SPoffset</td>
<td>offset value of SPstart in each symbol</td>
<td>3</td>
<td>int</td>
<td>[0, ∞)††</td>
</tr>
<tr>
<td>SPPhase</td>
<td>initial phase of scattered pilots</td>
<td>3</td>
<td>int</td>
<td>[0, SPPeriod/SPoff set-1)††††</td>
</tr>
</tbody>
</table>

† SPNumber = 9 for mode 1, 18 for mode 2, 36 for mode 3, per segment in ISDB-T systems.
†† SPPeriod = 12, SPStart = 0, SPoffset = 3 in ISDB-T systems.
††† SPPhase=3 in the OFDM receiver (because DTV_MLEstimator makes one OFDM symbol delay)

### Pin Inputs

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>input</td>
<td>received equalized signal</td>
<td>complex</td>
</tr>
</tbody>
</table>

4-20  DTV_DemuxCohSegs
1. The model is used to demultiplex the coherent modulation OFDM segments (such as QPSK, 16-QAM, and 64-QAM modulation) into TSP (transport stream packet) data, TMCC (transmission and multiplexing configuration control) data, AC (auxiliary channel) data according to Figure 4-4 and Table 4-5.

2. IntState of Carriers determines the number of TMCC and AC in each segment. Then, IntState of Start_Seg and Segments determines the TMCC and AC positions in each corresponding segment according to Table 4-5.

The PRBS sequences in each segment are generated according to Figure 4-5 and the initial sets of PRBS register in Table 4-6. The PRBS in initialized so that the first output bit from the PRBS coincides with the first active carrier. A new value is generated by the PRBS on every used carrier in each segment (whether or not it is a pilot).

The positions of the corresponding scattered pilots are generated as follows. For the symbol of index $l$ (ranging from 0 to 203), carriers for which index $k$ belongs to subset

$$\{ k \mid 3 \times l \mod 4 + 12 p \geq 0, \ k \geq 0, \ k \in \{0, \text{Length} \times \text{Segments}\} \}$$

are scattered pilots positions. Five parameters control the scattered pilots positions. SPnumber determines the number of scattered pilots in each segments, it is 9 in mode 1 (Length=108), 18 in mode 2 (Length=216), and 36 in mode 3 (Length=432). SPperiod=12, SPstart=0, SPoffset=3 and SPphase = 0 in all the three modes according to ISDB-T.

After determining TMCC, AC, and SP positions in each corresponding segment and the value of the PRBS sequence in all the active carriers in each segment, the model demultiplexes the input data into TMCC, AC, and TSP data. According to the TMCC position and the PRBS sequence, we can get the only one TMCC bit in each TMCC position, then the one transmitted TMCC bit is determined by the mean value of the TMCC positions.

---

**Pin Outputs**

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>output</td>
<td>TSP data output</td>
<td>complex</td>
</tr>
<tr>
<td>3</td>
<td>TMCC</td>
<td>TMCC data output</td>
<td>complex</td>
</tr>
<tr>
<td>4</td>
<td>AC</td>
<td>AC data output</td>
<td>complex</td>
</tr>
</tbody>
</table>

---

**Notes/Equations**

1. The model is used to demultiplex the coherent modulation OFDM segments (such as QPSK, 16-QAM, and 64-QAM modulation) into TSP (transport stream packet) data, TMCC (transmission and multiplexing configuration control) data, AC (auxiliary channel) data according to Figure 4-4 and Table 4-5.

2. IntState of Carriers determines the number of TMCC and AC in each segment. Then, IntState of Start_Seg and Segments determines the TMCC and AC positions in each corresponding segment according to Table 4-5.

The PRBS sequences in each segment are generated according to Figure 4-5 and the initial sets of PRBS register in Table 4-6. The PRBS in initialized so that the first output bit from the PRBS coincides with the first active carrier. A new value is generated by the PRBS on every used carrier in each segment (whether or not it is a pilot).

The positions of the corresponding scattered pilots are generated as follows. For the symbol of index $l$ (ranging from 0 to 203), carriers for which index $k$ belongs to subset

$$\{ k \mid 3 \times l \mod 4 + 12 p \geq 0, \ k \geq 0, \ k \in \{0, \text{Length} \times \text{Segments}\} \}$$

are scattered pilots positions. Five parameters control the scattered pilots positions. SPnumber determines the number of scattered pilots in each segments, it is 9 in mode 1 (Length=108), 18 in mode 2 (Length=216), and 36 in mode 3 (Length=432). SPperiod=12, SPstart=0, SPoffset=3 and SPphase = 0 in all the three modes according to ISDB-T.

After determining TMCC, AC, and SP positions in each corresponding segment and the value of the PRBS sequence in all the active carriers in each segment, the model demultiplexes the input data into TMCC, AC, and TSP data. According to the TMCC position and the PRBS sequence, we can get the only one TMCC bit in each TMCC position, then the one transmitted TMCC bit is determined by the mean value of the TMCC positions.
ISDB-T Components

\[
TMCC = \frac{1}{NTMCC} \sum_{l=1}^{NTMCC} \frac{x[TMCCposition[l]]}{PilotValue[TMCCposition[l]]}
\]

where NTMCC is the number of TMCCs in one segment; PilotValue is the PRBS sequence in corresponding segment; \(x[i]\) is the input data.

AC data is:

\[
AC[i] = \frac{x[ACposition[i]]}{PilotValue[ACposition[i]]}
\]

Except for AC, TMCC, and SP positions, the remaining positions in one segment are the TSP data positions.

Data[i] = x[i]
Figure 4-4. Structure of OFDM Segment for Coherent Modulation

Figure 4-5. Generation of PRBS Sequence
### ISDB-T Components

#### Table 4-5. Carrier Allocation of AC and TMCC for Coherent Modulation

<table>
<thead>
<tr>
<th>Segment</th>
<th>11</th>
<th>9</th>
<th>7</th>
<th>5</th>
<th>3</th>
<th>1</th>
<th>0</th>
<th>2</th>
<th>4</th>
<th>6</th>
<th>8</th>
<th>10</th>
<th>12</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Mode 1</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AC1_1</td>
<td>10</td>
<td>53</td>
<td>61</td>
<td>11</td>
<td>20</td>
<td>74</td>
<td>35</td>
<td>76</td>
<td>4</td>
<td>40</td>
<td>8</td>
<td>7</td>
<td>98</td>
</tr>
<tr>
<td>AC1_2</td>
<td>28</td>
<td>83</td>
<td>100</td>
<td>101</td>
<td>40</td>
<td>100</td>
<td>79</td>
<td>97</td>
<td>89</td>
<td>89</td>
<td>64</td>
<td>89</td>
<td>101</td>
</tr>
<tr>
<td>TMCC_1</td>
<td>70</td>
<td>25</td>
<td>17</td>
<td>86</td>
<td>44</td>
<td>47</td>
<td>49</td>
<td>31</td>
<td>83</td>
<td>61</td>
<td>85</td>
<td>101</td>
<td>23</td>
</tr>
<tr>
<td><strong>Mode 2</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AC1_1</td>
<td>10</td>
<td>61</td>
<td>20</td>
<td>35</td>
<td>4</td>
<td>8</td>
<td>98</td>
<td>53</td>
<td>11</td>
<td>74</td>
<td>76</td>
<td>40</td>
<td>7</td>
</tr>
<tr>
<td>AC1_2</td>
<td>28</td>
<td>100</td>
<td>40</td>
<td>79</td>
<td>89</td>
<td>64</td>
<td>101</td>
<td>83</td>
<td>101</td>
<td>100</td>
<td>97</td>
<td>89</td>
<td>89</td>
</tr>
<tr>
<td>AC1_3</td>
<td>161</td>
<td>119</td>
<td>182</td>
<td>184</td>
<td>148</td>
<td>115</td>
<td>118</td>
<td>169</td>
<td>128</td>
<td>143</td>
<td>112</td>
<td>116</td>
<td>206</td>
</tr>
<tr>
<td>AC1_4</td>
<td>191</td>
<td>209</td>
<td>208</td>
<td>205</td>
<td>197</td>
<td>197</td>
<td>136</td>
<td>208</td>
<td>148</td>
<td>187</td>
<td>197</td>
<td>172</td>
<td>209</td>
</tr>
<tr>
<td>TMCC_1</td>
<td>70</td>
<td>17</td>
<td>44</td>
<td>49</td>
<td>83</td>
<td>85</td>
<td>23</td>
<td>25</td>
<td>86</td>
<td>47</td>
<td>31</td>
<td>61</td>
<td>101</td>
</tr>
<tr>
<td>TMCC_2</td>
<td>133</td>
<td>194</td>
<td>155</td>
<td>139</td>
<td>169</td>
<td>209</td>
<td>178</td>
<td>125</td>
<td>152</td>
<td>157</td>
<td>191</td>
<td>193</td>
<td>131</td>
</tr>
<tr>
<td><strong>Mode 3</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AC1_1</td>
<td>10</td>
<td>20</td>
<td>4</td>
<td>98</td>
<td>11</td>
<td>76</td>
<td>7</td>
<td>61</td>
<td>35</td>
<td>8</td>
<td>53</td>
<td>74</td>
<td>40</td>
</tr>
<tr>
<td>AC1_2</td>
<td>28</td>
<td>40</td>
<td>89</td>
<td>101</td>
<td>101</td>
<td>97</td>
<td>89</td>
<td>100</td>
<td>79</td>
<td>64</td>
<td>83</td>
<td>100</td>
<td>89</td>
</tr>
<tr>
<td>AC1_3</td>
<td>161</td>
<td>182</td>
<td>148</td>
<td>118</td>
<td>128</td>
<td>112</td>
<td>206</td>
<td>119</td>
<td>184</td>
<td>115</td>
<td>169</td>
<td>143</td>
<td>116</td>
</tr>
<tr>
<td>AC1_4</td>
<td>191</td>
<td>208</td>
<td>197</td>
<td>136</td>
<td>148</td>
<td>197</td>
<td>209</td>
<td>209</td>
<td>205</td>
<td>197</td>
<td>208</td>
<td>187</td>
<td>172</td>
</tr>
<tr>
<td>AC1_5</td>
<td>277</td>
<td>251</td>
<td>224</td>
<td>269</td>
<td>290</td>
<td>256</td>
<td>226</td>
<td>236</td>
<td>220</td>
<td>314</td>
<td>227</td>
<td>292</td>
<td>223</td>
</tr>
<tr>
<td>AC1_6</td>
<td>316</td>
<td>295</td>
<td>280</td>
<td>299</td>
<td>316</td>
<td>305</td>
<td>244</td>
<td>256</td>
<td>305</td>
<td>317</td>
<td>317</td>
<td>313</td>
<td>305</td>
</tr>
<tr>
<td>AC1_7</td>
<td>335</td>
<td>400</td>
<td>331</td>
<td>385</td>
<td>359</td>
<td>332</td>
<td>377</td>
<td>398</td>
<td>364</td>
<td>334</td>
<td>342</td>
<td>328</td>
<td>422</td>
</tr>
<tr>
<td>AC1_8</td>
<td>425</td>
<td>421</td>
<td>413</td>
<td>424</td>
<td>403</td>
<td>388</td>
<td>407</td>
<td>424</td>
<td>413</td>
<td>352</td>
<td>364</td>
<td>413</td>
<td>425</td>
</tr>
<tr>
<td>TMCC_1</td>
<td>70</td>
<td>44</td>
<td>83</td>
<td>23</td>
<td>86</td>
<td>31</td>
<td>101</td>
<td>17</td>
<td>49</td>
<td>85</td>
<td>25</td>
<td>47</td>
<td>61</td>
</tr>
<tr>
<td>TMCC_2</td>
<td>131</td>
<td>155</td>
<td>169</td>
<td>178</td>
<td>152</td>
<td>191</td>
<td>131</td>
<td>194</td>
<td>139</td>
<td>209</td>
<td>125</td>
<td>157</td>
<td>193</td>
</tr>
<tr>
<td>TMCC_3</td>
<td>233</td>
<td>265</td>
<td>301</td>
<td>241</td>
<td>263</td>
<td>277</td>
<td>286</td>
<td>260</td>
<td>299</td>
<td>239</td>
<td>302</td>
<td>247</td>
<td>317</td>
</tr>
<tr>
<td>TMCC_4</td>
<td>410</td>
<td>355</td>
<td>425</td>
<td>341</td>
<td>373</td>
<td>409</td>
<td>349</td>
<td>371</td>
<td>385</td>
<td>394</td>
<td>368</td>
<td>407</td>
<td>347</td>
</tr>
</tbody>
</table>

#### Table 4-6. Initial Sets of PRBS Register

<table>
<thead>
<tr>
<th>Segment</th>
<th>Initial Sets for Mode 1†</th>
<th>Initial Sets for Mode 2‡</th>
<th>Initial Sets for Mode 3‡</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>1 1 1 1 1 1 1 1 1 1 1 1 1 1</td>
<td>1 1 1 1 1 1 1 1 1 1 1</td>
<td>1 1 1 1 1 1 1 1 1 1</td>
</tr>
<tr>
<td>9</td>
<td>1 1 0 1 0 0 1 1 1 1 1</td>
<td>0 1 1 0 1 0 1 1 1 1</td>
<td>1 1 0 1 1 0 0 1 1 1</td>
</tr>
<tr>
<td>7</td>
<td>0 1 1 0 1 0 1 1 1 1 0 1</td>
<td>1 1 0 1 1 0 0 1 0 1</td>
<td>1 0 0 1 0 0 0 0 0 0</td>
</tr>
<tr>
<td>5</td>
<td>0 1 0 0 0 1 0 1 1 1 0 1</td>
<td>1 1 0 0 1 0 0 0 1 0 1</td>
<td>0 1 1 1 0 0 1 1 1 0 1</td>
</tr>
<tr>
<td>3</td>
<td>1 1 0 1 1 0 0 1 0 1</td>
<td>1 0 1 0 1 0 0 0 0 0 0</td>
<td>0 0 1 0 0 1 1 0 1 1 0</td>
</tr>
</tbody>
</table>

† Initial Sets for Mode 1
‡ Initial Sets for Mode 2
§ Initial Sets for Mode 3
Table 4-6. Initial Sets of PRBS Register

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>001011110110</td>
<td>00001011000</td>
<td>11100110110</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>110010000010</td>
<td>01110001001</td>
<td>00100001011</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>000100001000</td>
<td>00000100100</td>
<td>11100111101</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>10010100000</td>
<td>00100011001</td>
<td>01101010011</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>11110110000</td>
<td>01100111001</td>
<td>10111010010</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>000001011000</td>
<td>11100110110</td>
<td>01100010010</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>10100100111</td>
<td>00101010001</td>
<td>01110100101</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>01110001001</td>
<td>00100001011</td>
<td>00100011100</td>
<td></td>
</tr>
</tbody>
</table>

†Degree from 0 to 10 in Figure 4-5.
ISDB-T Components

**DTV_DemuxDiffSegs**

Description  OFDM de-segment for differential modulation
Library   DTV, ISDB-T
Class   SDFDTV_DemuxDiffSegs

### Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Default</th>
<th>Type</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Carriers</td>
<td>number of carriers in one segment</td>
<td>432</td>
<td>int</td>
<td>108 for mode 1; 216 for mode 2; 432 for mode 3</td>
</tr>
<tr>
<td>Segments</td>
<td>number of segments</td>
<td>1</td>
<td>int</td>
<td>[1, 13]</td>
</tr>
<tr>
<td>Start_Seg</td>
<td>initial number of segment (0 to Segments-1)</td>
<td>0</td>
<td>int</td>
<td>[0, 12]</td>
</tr>
</tbody>
</table>

### Pin Inputs

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>input</td>
<td>received equalized signal</td>
<td>complex</td>
</tr>
</tbody>
</table>

### Pin Outputs

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>output</td>
<td>TSP data output</td>
<td>complex</td>
</tr>
<tr>
<td>3</td>
<td>TMCC</td>
<td>TMCC data output</td>
<td>complex</td>
</tr>
<tr>
<td>4</td>
<td>AC1</td>
<td>AC1 data output</td>
<td>complex</td>
</tr>
<tr>
<td>5</td>
<td>AC2</td>
<td>AC2 data output</td>
<td>complex</td>
</tr>
</tbody>
</table>

### Notes/Equations

1. The model is used to demultiplex the differential modulation OFDM segments (such as DQPSK modulation) into TSP (transport stream packet) data, TMCC
(transmission and multiplexing configuration control) data, AC (auxiliary channel) data according to Figure 4-6 and Table 4-7.

2. IntState of Carriers determines the number of TMCC, AC1, and AC2 in each segment. IntState of Start_Seg and Segments determine the TMCC, AC1, and AC2 positions in each corresponding segment according to Table 4-7.

The PRBS sequences in each segment are generated according to Figure 4-7 and the initial sets of PRBS register in Table 4-8. The PRBS in initialized so that the first output bit from the PRBS coincides with the first active carrier. A new value is generated by the PRBS on every used carriers in each segments (whether or not it is a pilot).

After determining all TMCC, AC1, and AC2 positions in each corresponding segment and the value of the PRBS sequence in all active carriers in each segment, the model demultiplexes the input data into TMCC, AC1, AC2, and TSP data. According to the TMCC position and the PRBS sequence, we can get the only one TMCC bit in each TMCC position, then the one transmitted TMCC bit is determined by the mean value of those TMCC positions.

\[
TMCC = \frac{1}{NTMCC} \sum_{l=1}^{NTMCC} \frac{x[TMCC_{position[i]}]}{PilotValue[TMCC_{position[i]}]}
\]

where NTMCC is the number of TMCCs in one segment; PilotValue is the PRBS sequence in corresponding segment; \(x[i]\) is the input data.

AC1 data is:

\[
AC1[i] = \frac{x[AC1_{position[i]}]}{PilotValue[AC1_{position[i]}]}
\]

AC2 data is:

\[
AC2[i] = \frac{x[AC2_{position[i]}]}{PilotValue[AC2_{position[i]}]}
\]

Except for the AC1, AC2 and TMCC data positions, the remaining positions in one segment are the TSP data positions. \(Data[i] = x[i]\).
ISDB-T Components

**Figure 4-6. Structure of OFDM Segment for Differential Modulation**

![Diagram showing OFDM segment structure](image)

\[ S_{i,j,k} \] denotes complex data in the data segment after time and frequency interleaving.

\( Nc = 108 \) for Mode 1; 216 for Mode 2; 432 for Mode 3

**Figure 4-7. Generation of PRBS Sequence**

![Diagram showing PRBS sequence generation](image)

\[ g(x) = x^{17} + x^9 + 1 \]

Output = \( W_i \)
Table 4-7. Carrier Allocation of CP, AC, and TMCC for Differential Modulation

<table>
<thead>
<tr>
<th>Segment</th>
<th>11</th>
<th>9</th>
<th>7</th>
<th>5</th>
<th>3</th>
<th>1</th>
<th>0</th>
<th>2</th>
<th>4</th>
<th>6</th>
<th>8</th>
<th>10</th>
<th>12</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Mode 1</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CP</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>AC1_1</td>
<td>10</td>
<td>53</td>
<td>61</td>
<td>11</td>
<td>20</td>
<td>74</td>
<td>35</td>
<td>76</td>
<td>4</td>
<td>40</td>
<td>8</td>
<td>7</td>
<td>98</td>
</tr>
<tr>
<td>AC1_2</td>
<td>28</td>
<td>83</td>
<td>100</td>
<td>101</td>
<td>40</td>
<td>100</td>
<td>79</td>
<td>97</td>
<td>89</td>
<td>89</td>
<td>64</td>
<td>89</td>
<td>101</td>
</tr>
<tr>
<td>AC2_1</td>
<td>3</td>
<td>3</td>
<td>29</td>
<td>28</td>
<td>23</td>
<td>30</td>
<td>3</td>
<td>5</td>
<td>13</td>
<td>72</td>
<td>36</td>
<td>25</td>
<td>10</td>
</tr>
<tr>
<td>AC2_2</td>
<td>45</td>
<td>15</td>
<td>41</td>
<td>45</td>
<td>63</td>
<td>81</td>
<td>72</td>
<td>18</td>
<td>93</td>
<td>95</td>
<td>48</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>AC2_3</td>
<td>59</td>
<td>40</td>
<td>84</td>
<td>81</td>
<td>85</td>
<td>92</td>
<td>85</td>
<td>57</td>
<td>98</td>
<td>100</td>
<td>52</td>
<td>42</td>
<td>55</td>
</tr>
<tr>
<td>AC2_4</td>
<td>77</td>
<td>58</td>
<td>93</td>
<td>91</td>
<td>105</td>
<td>103</td>
<td>89</td>
<td>92</td>
<td>102</td>
<td>105</td>
<td>74</td>
<td>104</td>
<td>81</td>
</tr>
<tr>
<td>TMCC_1</td>
<td>13</td>
<td>25</td>
<td>4</td>
<td>36</td>
<td>10</td>
<td>7</td>
<td>49</td>
<td>31</td>
<td>16</td>
<td>5</td>
<td>78</td>
<td>34</td>
<td>23</td>
</tr>
<tr>
<td>TMCC_2</td>
<td>50</td>
<td>63</td>
<td>7</td>
<td>48</td>
<td>28</td>
<td>25</td>
<td>61</td>
<td>39</td>
<td>30</td>
<td>10</td>
<td>82</td>
<td>48</td>
<td>37</td>
</tr>
<tr>
<td>TMCC_3</td>
<td>70</td>
<td>73</td>
<td>17</td>
<td>55</td>
<td>44</td>
<td>47</td>
<td>96</td>
<td>47</td>
<td>37</td>
<td>21</td>
<td>85</td>
<td>54</td>
<td>51</td>
</tr>
<tr>
<td>TMCC_4</td>
<td>83</td>
<td>80</td>
<td>51</td>
<td>59</td>
<td>47</td>
<td>60</td>
<td>99</td>
<td>65</td>
<td>74</td>
<td>44</td>
<td>98</td>
<td>70</td>
<td>68</td>
</tr>
<tr>
<td>TMCC_5</td>
<td>87</td>
<td>93</td>
<td>71</td>
<td>86</td>
<td>54</td>
<td>87</td>
<td>104</td>
<td>72</td>
<td>83</td>
<td>61</td>
<td>102</td>
<td>101</td>
<td>105</td>
</tr>
<tr>
<td><strong>Mode 2</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CP</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>AC1_1</td>
<td>10</td>
<td>61</td>
<td>20</td>
<td>35</td>
<td>4</td>
<td>8</td>
<td>98</td>
<td>53</td>
<td>11</td>
<td>74</td>
<td>76</td>
<td>40</td>
<td>7</td>
</tr>
<tr>
<td>AC1_2</td>
<td>28</td>
<td>100</td>
<td>40</td>
<td>79</td>
<td>89</td>
<td>64</td>
<td>101</td>
<td>83</td>
<td>101</td>
<td>100</td>
<td>97</td>
<td>89</td>
<td>89</td>
</tr>
<tr>
<td>AC1_3</td>
<td>161</td>
<td>119</td>
<td>182</td>
<td>184</td>
<td>148</td>
<td>115</td>
<td>118</td>
<td>169</td>
<td>128</td>
<td>143</td>
<td>112</td>
<td>116</td>
<td>206</td>
</tr>
<tr>
<td>AC1_4</td>
<td>191</td>
<td>209</td>
<td>205</td>
<td>197</td>
<td>197</td>
<td>136</td>
<td>208</td>
<td>148</td>
<td>187</td>
<td>197</td>
<td>172</td>
<td>209</td>
<td></td>
</tr>
<tr>
<td>AC2_1</td>
<td>3</td>
<td>29</td>
<td>23</td>
<td>3</td>
<td>13</td>
<td>36</td>
<td>10</td>
<td>3</td>
<td>28</td>
<td>30</td>
<td>5</td>
<td>72</td>
<td>25</td>
</tr>
<tr>
<td>AC2_2</td>
<td>45</td>
<td>41</td>
<td>63</td>
<td>72</td>
<td>93</td>
<td>48</td>
<td>30</td>
<td>15</td>
<td>45</td>
<td>81</td>
<td>18</td>
<td>95</td>
<td>30</td>
</tr>
<tr>
<td>AC2_3</td>
<td>59</td>
<td>84</td>
<td>85</td>
<td>85</td>
<td>98</td>
<td>52</td>
<td>55</td>
<td>40</td>
<td>81</td>
<td>92</td>
<td>57</td>
<td>100</td>
<td>42</td>
</tr>
<tr>
<td>AC2_4</td>
<td>77</td>
<td>93</td>
<td>105</td>
<td>89</td>
<td>102</td>
<td>74</td>
<td>81</td>
<td>58</td>
<td>91</td>
<td>103</td>
<td>92</td>
<td>105</td>
<td>104</td>
</tr>
<tr>
<td>AC2_6</td>
<td>113</td>
<td>136</td>
<td>136</td>
<td>136</td>
<td>136</td>
<td>136</td>
<td>136</td>
<td>136</td>
<td>136</td>
<td>136</td>
<td>136</td>
<td>136</td>
<td>136</td>
</tr>
<tr>
<td>AC2_7</td>
<td>123</td>
<td>153</td>
<td>189</td>
<td>126</td>
<td>203</td>
<td>138</td>
<td>153</td>
<td>149</td>
<td>171</td>
<td>180</td>
<td>201</td>
<td>156</td>
<td>138</td>
</tr>
<tr>
<td>AC2_8</td>
<td>148</td>
<td>189</td>
<td>200</td>
<td>165</td>
<td>208</td>
<td>150</td>
<td>167</td>
<td>192</td>
<td>193</td>
<td>193</td>
<td>206</td>
<td>160</td>
<td>163</td>
</tr>
<tr>
<td>AC2_9</td>
<td>166</td>
<td>199</td>
<td>211</td>
<td>200</td>
<td>213</td>
<td>212</td>
<td>185</td>
<td>201</td>
<td>213</td>
<td>197</td>
<td>210</td>
<td>182</td>
<td>189</td>
</tr>
<tr>
<td>TMCC_1</td>
<td>13</td>
<td>4</td>
<td>10</td>
<td>49</td>
<td>16</td>
<td>76</td>
<td>23</td>
<td>25</td>
<td>36</td>
<td>7</td>
<td>31</td>
<td>5</td>
<td>34</td>
</tr>
<tr>
<td>TMCC_2</td>
<td>50</td>
<td>7</td>
<td>28</td>
<td>61</td>
<td>30</td>
<td>82</td>
<td>37</td>
<td>63</td>
<td>48</td>
<td>25</td>
<td>39</td>
<td>10</td>
<td>48</td>
</tr>
<tr>
<td>TMCC_3</td>
<td>70</td>
<td>17</td>
<td>44</td>
<td>96</td>
<td>37</td>
<td>85</td>
<td>51</td>
<td>73</td>
<td>55</td>
<td>47</td>
<td>47</td>
<td>21</td>
<td>54</td>
</tr>
<tr>
<td>TMCC_4</td>
<td>83</td>
<td>51</td>
<td>47</td>
<td>99</td>
<td>74</td>
<td>98</td>
<td>68</td>
<td>80</td>
<td>59</td>
<td>60</td>
<td>65</td>
<td>44</td>
<td>70</td>
</tr>
<tr>
<td>TMCC_5</td>
<td>87</td>
<td>71</td>
<td>54</td>
<td>104</td>
<td>83</td>
<td>102</td>
<td>105</td>
<td>93</td>
<td>86</td>
<td>87</td>
<td>72</td>
<td>61</td>
<td>101</td>
</tr>
</tbody>
</table>
### Table 4-7. Carrier Allocation of CP, AC, and TMCC for Differential Modulation (continued)

<table>
<thead>
<tr>
<th>Segment</th>
<th>1</th>
<th>11</th>
<th>9</th>
<th>7</th>
<th>5</th>
<th>3</th>
<th>1</th>
<th>0</th>
<th>2</th>
<th>4</th>
<th>6</th>
<th>8</th>
<th>10</th>
<th>12</th>
</tr>
</thead>
<tbody>
<tr>
<td>TMCC_6</td>
<td>133</td>
<td>144</td>
<td>115</td>
<td>139</td>
<td>113</td>
<td>142</td>
<td>121</td>
<td>112</td>
<td>118</td>
<td>157</td>
<td>124</td>
<td>186</td>
<td>131</td>
<td></td>
</tr>
<tr>
<td>TMCC_7</td>
<td>171</td>
<td>156</td>
<td>133</td>
<td>147</td>
<td>118</td>
<td>156</td>
<td>158</td>
<td>115</td>
<td>136</td>
<td>169</td>
<td>138</td>
<td>190</td>
<td>145</td>
<td></td>
</tr>
<tr>
<td>TMCC_8</td>
<td>181</td>
<td>163</td>
<td>155</td>
<td>155</td>
<td>129</td>
<td>162</td>
<td>178</td>
<td>125</td>
<td>152</td>
<td>204</td>
<td>145</td>
<td>193</td>
<td>159</td>
<td></td>
</tr>
<tr>
<td>TMCC_9</td>
<td>188</td>
<td>167</td>
<td>168</td>
<td>173</td>
<td>152</td>
<td>178</td>
<td>191</td>
<td>159</td>
<td>155</td>
<td>207</td>
<td>182</td>
<td>206</td>
<td>176</td>
<td></td>
</tr>
<tr>
<td>TMCC_10</td>
<td>201</td>
<td>194</td>
<td>195</td>
<td>180</td>
<td>169</td>
<td>209</td>
<td>195</td>
<td>179</td>
<td>162</td>
<td>212</td>
<td>191</td>
<td>210</td>
<td>213</td>
<td></td>
</tr>
<tr>
<td>Mode 3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CP</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AC1_1</td>
<td>10</td>
<td>20</td>
<td>4</td>
<td>98</td>
<td>11</td>
<td>76</td>
<td>7</td>
<td>61</td>
<td>35</td>
<td>8</td>
<td>53</td>
<td>74</td>
<td>40</td>
<td></td>
</tr>
<tr>
<td>AC1_2</td>
<td>28</td>
<td>40</td>
<td>89</td>
<td>101</td>
<td>101</td>
<td>97</td>
<td>89</td>
<td>100</td>
<td>79</td>
<td>64</td>
<td>83</td>
<td>100</td>
<td>89</td>
<td></td>
</tr>
<tr>
<td>AC1_3</td>
<td>161</td>
<td>182</td>
<td>148</td>
<td>118</td>
<td>128</td>
<td>112</td>
<td>206</td>
<td>119</td>
<td>184</td>
<td>115</td>
<td>169</td>
<td>143</td>
<td>116</td>
<td></td>
</tr>
<tr>
<td>AC1_4</td>
<td>191</td>
<td>208</td>
<td>197</td>
<td>136</td>
<td>148</td>
<td>197</td>
<td>209</td>
<td>209</td>
<td>205</td>
<td>197</td>
<td>208</td>
<td>187</td>
<td>172</td>
<td></td>
</tr>
<tr>
<td>AC1_5</td>
<td>277</td>
<td>251</td>
<td>224</td>
<td>269</td>
<td>290</td>
<td>256</td>
<td>226</td>
<td>236</td>
<td>220</td>
<td>314</td>
<td>227</td>
<td>292</td>
<td>223</td>
<td></td>
</tr>
<tr>
<td>AC1_6</td>
<td>316</td>
<td>295</td>
<td>280</td>
<td>299</td>
<td>316</td>
<td>305</td>
<td>244</td>
<td>256</td>
<td>305</td>
<td>317</td>
<td>317</td>
<td>313</td>
<td>305</td>
<td></td>
</tr>
<tr>
<td>AC1_7</td>
<td>335</td>
<td>400</td>
<td>331</td>
<td>385</td>
<td>359</td>
<td>332</td>
<td>377</td>
<td>398</td>
<td>364</td>
<td>334</td>
<td>344</td>
<td>328</td>
<td>422</td>
<td></td>
</tr>
<tr>
<td>AC1_8</td>
<td>425</td>
<td>421</td>
<td>413</td>
<td>424</td>
<td>403</td>
<td>388</td>
<td>407</td>
<td>424</td>
<td>413</td>
<td>352</td>
<td>364</td>
<td>413</td>
<td>425</td>
<td></td>
</tr>
<tr>
<td>AC2_1</td>
<td>3</td>
<td>23</td>
<td>13</td>
<td>10</td>
<td>28</td>
<td>5</td>
<td>25</td>
<td>29</td>
<td>3</td>
<td>36</td>
<td>3</td>
<td>30</td>
<td>72</td>
<td></td>
</tr>
<tr>
<td>AC2_2</td>
<td>45</td>
<td>63</td>
<td>93</td>
<td>30</td>
<td>45</td>
<td>18</td>
<td>30</td>
<td>41</td>
<td>72</td>
<td>48</td>
<td>15</td>
<td>81</td>
<td>95</td>
<td></td>
</tr>
<tr>
<td>AC2_3</td>
<td>59</td>
<td>85</td>
<td>98</td>
<td>55</td>
<td>81</td>
<td>57</td>
<td>42</td>
<td>84</td>
<td>85</td>
<td>52</td>
<td>40</td>
<td>92</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>AC2_4</td>
<td>77</td>
<td>105</td>
<td>102</td>
<td>81</td>
<td>91</td>
<td>92</td>
<td>104</td>
<td>93</td>
<td>89</td>
<td>74</td>
<td>58</td>
<td>103</td>
<td>105</td>
<td></td>
</tr>
<tr>
<td>AC2_6</td>
<td>111</td>
<td>138</td>
<td>180</td>
<td>111</td>
<td>131</td>
<td>121</td>
<td>118</td>
<td>136</td>
<td>113</td>
<td>133</td>
<td>137</td>
<td>111</td>
<td>144</td>
<td></td>
</tr>
<tr>
<td>AC2_7</td>
<td>123</td>
<td>189</td>
<td>203</td>
<td>153</td>
<td>171</td>
<td>201</td>
<td>138</td>
<td>153</td>
<td>126</td>
<td>138</td>
<td>149</td>
<td>180</td>
<td>156</td>
<td></td>
</tr>
<tr>
<td>AC2_8</td>
<td>148</td>
<td>200</td>
<td>208</td>
<td>167</td>
<td>193</td>
<td>206</td>
<td>163</td>
<td>189</td>
<td>165</td>
<td>150</td>
<td>192</td>
<td>193</td>
<td>160</td>
<td></td>
</tr>
<tr>
<td>AC2_9</td>
<td>166</td>
<td>211</td>
<td>213</td>
<td>185</td>
<td>213</td>
<td>210</td>
<td>189</td>
<td>199</td>
<td>200</td>
<td>212</td>
<td>201</td>
<td>197</td>
<td>182</td>
<td></td>
</tr>
<tr>
<td>AC2_10</td>
<td>216</td>
<td>216</td>
<td>216</td>
<td>216</td>
<td>216</td>
<td>216</td>
<td>216</td>
<td>216</td>
<td>216</td>
<td>216</td>
<td>216</td>
<td>216</td>
<td>216</td>
<td></td>
</tr>
<tr>
<td>AC2_11</td>
<td>245</td>
<td>219</td>
<td>252</td>
<td>219</td>
<td>246</td>
<td>288</td>
<td>219</td>
<td>239</td>
<td>229</td>
<td>226</td>
<td>244</td>
<td>221</td>
<td>241</td>
<td></td>
</tr>
<tr>
<td>AC2_12</td>
<td>257</td>
<td>288</td>
<td>284</td>
<td>231</td>
<td>297</td>
<td>311</td>
<td>261</td>
<td>279</td>
<td>309</td>
<td>246</td>
<td>261</td>
<td>234</td>
<td>246</td>
<td></td>
</tr>
<tr>
<td>AC2_13</td>
<td>300</td>
<td>301</td>
<td>268</td>
<td>256</td>
<td>308</td>
<td>316</td>
<td>275</td>
<td>301</td>
<td>314</td>
<td>271</td>
<td>297</td>
<td>273</td>
<td>258</td>
<td></td>
</tr>
<tr>
<td>AC2_14</td>
<td>309</td>
<td>305</td>
<td>290</td>
<td>274</td>
<td>319</td>
<td>321</td>
<td>293</td>
<td>321</td>
<td>318</td>
<td>297</td>
<td>307</td>
<td>308</td>
<td>320</td>
<td></td>
</tr>
<tr>
<td>AC2_15</td>
<td>324</td>
<td>324</td>
<td>324</td>
<td>324</td>
<td>324</td>
<td>324</td>
<td>324</td>
<td>324</td>
<td>324</td>
<td>324</td>
<td>324</td>
<td>324</td>
<td>324</td>
<td></td>
</tr>
<tr>
<td>AC2_16</td>
<td>352</td>
<td>329</td>
<td>349</td>
<td>353</td>
<td>327</td>
<td>360</td>
<td>327</td>
<td>354</td>
<td>396</td>
<td>327</td>
<td>347</td>
<td>337</td>
<td>334</td>
<td></td>
</tr>
<tr>
<td>AC2_17</td>
<td>369</td>
<td>342</td>
<td>354</td>
<td>365</td>
<td>396</td>
<td>372</td>
<td>339</td>
<td>405</td>
<td>419</td>
<td>369</td>
<td>387</td>
<td>417</td>
<td>354</td>
<td></td>
</tr>
<tr>
<td>AC2_18</td>
<td>405</td>
<td>381</td>
<td>366</td>
<td>408</td>
<td>409</td>
<td>376</td>
<td>364</td>
<td>416</td>
<td>424</td>
<td>383</td>
<td>409</td>
<td>422</td>
<td>379</td>
<td></td>
</tr>
<tr>
<td>AC2_19</td>
<td>415</td>
<td>416</td>
<td>428</td>
<td>417</td>
<td>413</td>
<td>398</td>
<td>382</td>
<td>427</td>
<td>429</td>
<td>401</td>
<td>429</td>
<td>426</td>
<td>405</td>
<td></td>
</tr>
</tbody>
</table>
### Table 4-8. Initial Sets of PRBS Register

<table>
<thead>
<tr>
<th>Segment</th>
<th>Initial Sets for Mode 1</th>
<th>Initial Sets for Mode 2</th>
<th>Initial Sets for Mode 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>111111111111</td>
<td>111111111111</td>
<td>111111111111</td>
</tr>
<tr>
<td>9</td>
<td>110110011111</td>
<td>011010111110</td>
<td>110111001001</td>
</tr>
<tr>
<td>7</td>
<td>011010111110</td>
<td>110111001001</td>
<td>100101000000</td>
</tr>
<tr>
<td>5</td>
<td>010001011110</td>
<td>110010000010</td>
<td>011100010001</td>
</tr>
<tr>
<td>3</td>
<td>110111000101</td>
<td>100101000000</td>
<td>001000110001</td>
</tr>
<tr>
<td>1</td>
<td>001011100100</td>
<td>000010110000</td>
<td>111001100100</td>
</tr>
<tr>
<td>0</td>
<td>110010000000</td>
<td>011100010001</td>
<td>001000110001</td>
</tr>
<tr>
<td>2</td>
<td>000100000100</td>
<td>000001001000</td>
<td>111001110101</td>
</tr>
<tr>
<td>4</td>
<td>100101000000</td>
<td>001000110001</td>
<td>011010100011</td>
</tr>
<tr>
<td>6</td>
<td>111101100000</td>
<td>011001110001</td>
<td>101110100010</td>
</tr>
</tbody>
</table>
ISDB-T Components

Table 4-8. Initial Sets of PRBS Register

<table>
<thead>
<tr>
<th>8</th>
<th>00001011000</th>
<th>11100110110</th>
<th>01100010010</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>10100100111</td>
<td>00101010001</td>
<td>11010010010</td>
</tr>
<tr>
<td>12</td>
<td>01110001001</td>
<td>00100001011</td>
<td>00010011100</td>
</tr>
</tbody>
</table>

†Degree from 0 to 10 in Figure 4-7.

References

DTV_DemuxTMCC

Description  TMCC bit decomposer into 20 and 184 bits)
Library  DTV, ISDB-T
Class  SDFDTV_DemuxTMCC

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Default</th>
<th>Type</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Length</td>
<td>length of TMCC format</td>
<td>204</td>
<td>int</td>
<td>204</td>
</tr>
</tbody>
</table>

Pin Inputs

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>input</td>
<td>received equalized signal</td>
<td>int</td>
</tr>
</tbody>
</table>

Pin Outputs

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>Sync</td>
<td>synchronization of TMCC output</td>
<td>int</td>
</tr>
<tr>
<td>3</td>
<td>Desc</td>
<td>segment descriptor output</td>
<td>int</td>
</tr>
<tr>
<td>4</td>
<td>Info</td>
<td>coded TMCC information</td>
<td>int</td>
</tr>
</tbody>
</table>

Notes/Equations

1. The model is used to demultiplex the received 204 transmission and multiplexing configuration control bits in ISDB-T systems.

2. According to ISDB-T, the first bit $B_0$ is the initialization bit, so this bit is discarded.

   $B_1 - B_{16}$, a 16-bit synchronization sequence, takes $w_0$ and $w_1$ (the inverse of $w_0$) in turn in every frame. This model outputs this 16-bit synchronization sequence.
ISDB-T Components

$B_{17} - B_{19}$ represent the segment descriptor.

$B_{20} - B_{203}$ include TMCC information (102 bits) and 82-bit parity bits; these 184 bits need CDSC decoding.

References

DTV_InterSegInterlv

**Description**  Inter-segment interleaving of OFDM symbols

**Library**  DTV, ISDB-T

**Class**  SDFDTV_InterSegInterlv

**Parameters**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Default</th>
<th>Type</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Segments</td>
<td>depth of block interleaver</td>
<td>4</td>
<td>int</td>
<td>[1, 13]</td>
</tr>
<tr>
<td>Carriers</td>
<td>width of block interleaver</td>
<td>384</td>
<td>int</td>
<td>[96, 192, 384]</td>
</tr>
<tr>
<td>Option</td>
<td>operating option:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Interleaving,</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Deinterleaving:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Interleave, Deinterleave</td>
<td>Interleave</td>
<td>enum</td>
<td></td>
</tr>
</tbody>
</table>

**Pin Inputs**

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>input</td>
<td>input symbols to be interleaved</td>
<td>anytype</td>
</tr>
</tbody>
</table>

**Pin Outputs**

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>output</td>
<td>output symbols after interleaved</td>
<td>anytype</td>
</tr>
</tbody>
</table>

**Notes/Equations**

1. This model is used to perform inter-segment symbol interleaving among all segments within the differential or coherent modulation layer.

2. The inter-segment interleaving is carried out among differential modulation (DQPSK) segments and coherent modulation (QPSK, 16-QAM, and 64-QAM) as shown in Figure 4-8.
ISDB-T Components

Figure 4-8. Inter-Segment Interleaver

Allocation of Complex Data before Inter-Segment Interleaving

Allocation of Complex Data after Inter-Segment Interleaving

(a) Inter-Segment Interleaver for Mode 1

Allocation of Complex Data before Inter-Segment Interleaving

Allocation of Complex Data after Inter-Segment Interleaving

(b) Inter-Segment Interleaver for Mode 2

Allocation of Complex Data before Inter-Segment Interleaving

Allocation of Complex Data after Inter-Segment Interleaving

(c) Inter-Segment Interleaver for Mode 3

$S_{i,j,k}$ denotes complex data of OFDM segment.
References

ISDB-T Components

**DTV_LFSRCoder**

**Description**  
LFSR cyclic coder

**Library**  
DTV, ISDB-T

**Class**  
SDFDTV_LFSRCoder

**Parameters**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Default</th>
<th>Type</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Code</td>
<td>length of code bits</td>
<td>184</td>
<td>int</td>
<td>(0, ∞)</td>
</tr>
<tr>
<td>Info</td>
<td>length of information bits</td>
<td>102</td>
<td>int</td>
<td>(0, ∞)</td>
</tr>
<tr>
<td>Polynomial</td>
<td>generation polynomial's suffix (X^{P[0]} + X^{P[1]} + \ldots + X^{P[n]})</td>
<td>[0 4 10 18 22 24 34 36 40 48 52 56 66 67 71 76 77 82]</td>
<td>int array</td>
<td>[0, ∞)</td>
</tr>
</tbody>
</table>

**Pin Inputs**

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>input</td>
<td>signal to be encoded</td>
<td>int</td>
</tr>
</tbody>
</table>

**Pin Outputs**

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>output</td>
<td>error protected signal</td>
<td>int</td>
</tr>
</tbody>
</table>

**Notes/Equations**

1. This model is used to perform cyclic coding over the input signal.

2. Cyclic codes are linear block codes with an important additional property: a cyclic, or end-around, shift of any codeword is also a codeword. In the language of algebra, the codewords constitute a group under the cyclic shift operation.
If \( C = (c_0, c_1, \ldots, c_{n-1}) \) denotes a codeword with elements in \( \text{GF}(q) \), we associate with it a polynomial over \( \text{GF}(q) \) of degree at most \( n-1 \):

\[
c(x) = c_0 + c_1 x + c_2 x^2 + \ldots + c_{n-1} x^{n-1}
\]

Note the coefficients of the polynomial are in \( \text{GF}(q) \). Now consider a one-position right cyclic shift of \( C \), producing

\[
C^{(1)} = (c_{n-1}, c_0, \ldots, c_{n-2}).
\]

The associated polynomial would be

\[
c^{(1)}(x) = c_{n-1} + c_0 x + c_1 x^2 + \ldots + c_{n-2} x^{n-1}
\]

which is another polynomial of degree at most \( n-1 \).

Polynomials \( c(x) \) and \( c^{(1)}(x) \) are related by

\[
c^{(1)}(x) = xc(x) \mod (x^n - 1)
\]

Now suppose that we are given a particular cyclic \((n,k)\) code over \( \text{GF}(q) \). We define the generator polynomial \( g(x) \) of the cyclic code as the monic polynomial of minimum degree among the set of nonzero codeword polynomials. We suppose that the degree of this polynomial is \( r \leq n-1 \) and write

\[
g(x) = g_0 + g_1 x + g_2 x^2 + \ldots + g_{n-1} x^{n-1}
\]

where again the coefficients are members of \( \text{GF}(q) \).

Two fundamental properties of cyclic codes are:

- Property 1. \( c(x) \) is a code polynomial if \( c(x) = u(x)g(x) \), where \( u(x) \) is of degree \( n-1-r \) or less.

- Property 2. \( g(x) \) generates an \((n,k)\) cyclic code if \( g(x) \) is of degree \( n-k \) and is a factor of \( x^n - 1 \).

Property 1 of cyclic codes suggests one implementation of an encoder: performing the computation of \( c(x) = u(x)g(x) \). Such an encoding system is shown in Figure 4-9 and is nothing more than a digital transversal filter over \( \text{GF}(q) \) that convolves the information sequence \( u_{k-1}, \ldots, u_0 \) with the impulse response \( g_{n-k}, g_{n-k-1}, \ldots, g_0 \).
ISDB-T Components

![Diagram of Non-Systematic Encoder for Cyclic Code]

Figure 4-9. Non-Systematic Encoder for Cyclic Code

References


Description   Layer data stream loader into IFFT buffer with transmission spectrum adjustment for ISDB-T
Library   DTV, ISDB-T
Class   SDFDTV_LoadIFFTBuff

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Default</th>
<th>Type</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Carriers</td>
<td>number of carriers in one segment</td>
<td>432</td>
<td>int</td>
<td>108 for mode 1; 216 for mode 2; 432 for mode 3</td>
</tr>
<tr>
<td>Segments</td>
<td>number of segments</td>
<td>13</td>
<td>int</td>
<td>[1, 13]</td>
</tr>
<tr>
<td>InSequence</td>
<td>segment sequence at input</td>
<td>0 1 2 3 4 5 6 7 8 9</td>
<td>array</td>
<td>&quot;0, 1,...,Segments-1&quot;†</td>
</tr>
<tr>
<td>OutSequence</td>
<td>segment sequence at output</td>
<td>1 1 9 7 5 3 1 0 2 4 6 8 10 12</td>
<td>array</td>
<td>&quot;... 1,0,2,...&quot;††</td>
</tr>
<tr>
<td>Order</td>
<td>IFFT points=2^Order</td>
<td>13</td>
<td>int</td>
<td>[1, ∞)†††</td>
</tr>
</tbody>
</table>

† The InSequence value is [0, Segments-1]. According to ISDB-T, InSequence is "0, 1, 2,..., Segments-1" after Segments is determined.
†† The OutSequence value is [0, Segments-1]. According to ISDB-T, OutSequence is "...1,0,2,..." after Segments is determined. For example, if Segments=5, OutSequence is "3,1,0,2,4"; if Segments=10, OutSequence is "9,7,5,3,1,0,2,4,6,8".
††† Order is the order of FFT. It must satisfy 2^Order ≥ Carriers * Segments

Pin Inputs

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>input</td>
<td>received segments signal</td>
<td>complex</td>
</tr>
</tbody>
</table>
**Note Equations**

1. This model is used to load the adjusted data segments into the IFFT buffer with transmission spectrum adjustment for ISDB-T.

2. The input data segment order is adjusted in order to change its transmission spectrum. The input segment is changed into the output segment; the changed segments are placed in the IFFT buffer by zero padded in the center of the buffer.

   Assume $x(0), x(1), \ldots, x(N-1)$ are the signal of the changed segments, where

   \[
   N = \text{Length} \times \text{Segments} + 1
   \]

   \[
   M = 2^{\text{FFTStage}}
   \]

   $y(0), y(1), \ldots, y(M-1)$ are the outputs of the model. Data loading is performed as follows:

   \[
y(i) = x\left(\frac{N}{2} + i\right)
   \quad i = 0, \ldots, \frac{N + 1}{2} - 1
   \]

   \[
y(i) = 0
   \quad i = \frac{N + 1}{2}, \ldots, M - \frac{N}{2} - 1
   \]

   \[
y(i) = x\left(i - M + \frac{N}{2}\right)
   \quad i = M - \frac{N}{2}, \ldots, M
   \]

**References**

DTD_MuxCohSegs

Description ISDB-T multiplex coherent segments  
Library DTV, ISDB-T  
Class SDFDTV_MuxCohSegs

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Default</th>
<th>Type</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Carriers</td>
<td>number of carriers in one segment</td>
<td>432</td>
<td>int</td>
<td>108 for mode 1; 216 for mode 2; 432 for mode 3</td>
</tr>
<tr>
<td>Segments</td>
<td>number of segments</td>
<td>1</td>
<td>int</td>
<td>[1, 13]</td>
</tr>
<tr>
<td>Start_Seg</td>
<td>initial number of segment (0 to Segments-1)</td>
<td>0</td>
<td>int</td>
<td>[0, 12]</td>
</tr>
<tr>
<td>SPNumber</td>
<td>number of scattered pilots in each segment</td>
<td>36</td>
<td>int</td>
<td>[0, ∞]†</td>
</tr>
<tr>
<td>SPperiod</td>
<td>distance in carriers between nearby scattered pilots</td>
<td>12</td>
<td>int</td>
<td>(0, ∞)††</td>
</tr>
<tr>
<td>SPstart</td>
<td>start position of scattered pilots in carriers</td>
<td>0</td>
<td>int</td>
<td>(0, ∞)††</td>
</tr>
<tr>
<td>SPOffset</td>
<td>offset value of SPstart in each symbol</td>
<td>3</td>
<td>int</td>
<td>(0, ∞)††</td>
</tr>
<tr>
<td>SPPhase</td>
<td>initial phase of scattered pilots</td>
<td>0</td>
<td>int</td>
<td>[0, SPperiod/SPOff set-1]</td>
</tr>
</tbody>
</table>

† SPnumber = 9 for mode 1, 18 for mode 2, 36 for mode 3, per segment in ISDB-T systems.  
†† SPperiod = 12, SPstart = 0, SPOffset = 3 in ISDB-T systems.

Pin Inputs

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>data</td>
<td>TSP data input</td>
<td>complex</td>
</tr>
</tbody>
</table>
ISDB-T Components

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>TMCC</td>
<td>TMCC data input</td>
<td>complex</td>
</tr>
<tr>
<td>3</td>
<td>AC</td>
<td>AC data input</td>
<td>complex</td>
</tr>
</tbody>
</table>

**Pin Outputs**

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>output</td>
<td>coherent segments data</td>
<td>complex</td>
</tr>
</tbody>
</table>

**Notes/Equations**

1. The model is used to multiplex TSP (transport stream packet) data, TMCC (transmission and multiplexing configuration control) data, and AC (auxiliary channel) data into the coherent modulation OFDM segments (such as QPSK, 16-QAM, and 64-QAM modulation) according to Figure 4-10 and Table 4-9.

2. IntState of Carriers determines the number of TMCC and AC in each segment. Then, IntState of Start_Seg and Segments determines the TMCC and AC positions in each corresponding segment according to Table 4-9.

The PRBS sequences in each segment are generated according to Figure 4-11 and the initial sets of PRBS register in Table 4-10. The PRBS is initialized so that the first output bit from the PRBS coincides with the first active carrier. A new value is generated by the PRBS on every used carriers in each segment (whether or not it is a pilot).

Positions of corresponding scattered pilots are generated as follows. For symbol of index \( l \) (ranging from 0 to 203), carriers for which index \( k \) belongs to subset \( \{ k = 3 \times l \mod 4 + 12p | p \text{ integer, } p \geq 0, k \in [0, \text{Length} \times \text{Segments}] \} \)

are scattered pilots positions. Scattered pilots positions are controlled by

- SPnumber, which determines the number of scattered pilots in each segment, is: 9 in mode 1 (Length=108), 18 in mode 2 (Length=216), or 36 in mode 3 (Length=432).
- SP period=12, SPstart=0, SPOffset=3 in all three modes according to ISDB-T.

After determining TMCC, AC, and SP positions in each corresponding segment and the value of the PRBS sequence in all active carriers in each segment, TMCC, AC, and TSP data are multiplexed into the coherent segments.
According to the TMCC position and the PRBS sequence, one TMCC bit is output in every TMCC position.

\[ x[\text{TMCC position}[l]] = \text{PilotValue}[\text{TMCC position}[l]] \times \text{TMCC} \]

where NTMCC is the number of TMCCs in one segment; PilotValue is the PRBS sequence in the corresponding segment; \( x[i] \) is segment data.

AC data is:

\[ x[\text{AC position}[i]] = \text{PilotValue}[\text{AC position}[i]] \times \text{AC}[i] \]

Except for AC, TMCC, and SP positions, the remaining positions in one segment are TSP data positions.

\[ x[i] = \text{data}[i] \]
ISDB-T Components

Figure 4-10. Structure of OFDM Segment for Coherent Modulation

Figure 4-11. Generation of PRBS Sequence

\[ g(x) = x^7 + x^9 + 1 \]
### Table 4-9. Carrier Allocation of AC and TMCC for Coherent Modulation

<table>
<thead>
<tr>
<th>Segment</th>
<th>11</th>
<th>9</th>
<th>7</th>
<th>5</th>
<th>3</th>
<th>1</th>
<th>0</th>
<th>2</th>
<th>4</th>
<th>6</th>
<th>8</th>
<th>10</th>
<th>12</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Mode 1</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AC1_1</td>
<td>10</td>
<td>53</td>
<td>81</td>
<td>11</td>
<td>20</td>
<td>74</td>
<td>35</td>
<td>76</td>
<td>4</td>
<td>40</td>
<td>8</td>
<td>7</td>
<td>98</td>
</tr>
<tr>
<td>AC1_2</td>
<td>28</td>
<td>83</td>
<td>100</td>
<td>101</td>
<td>40</td>
<td>100</td>
<td>79</td>
<td>97</td>
<td>89</td>
<td>89</td>
<td>64</td>
<td>89</td>
<td>101</td>
</tr>
<tr>
<td>TMCC_1</td>
<td>70</td>
<td>25</td>
<td>17</td>
<td>86</td>
<td>44</td>
<td>47</td>
<td>49</td>
<td>31</td>
<td>83</td>
<td>61</td>
<td>85</td>
<td>101</td>
<td>23</td>
</tr>
<tr>
<td><strong>Mode 2</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AC1_1</td>
<td>10</td>
<td>61</td>
<td>20</td>
<td>35</td>
<td>4</td>
<td>8</td>
<td>98</td>
<td>53</td>
<td>11</td>
<td>74</td>
<td>76</td>
<td>40</td>
<td>7</td>
</tr>
<tr>
<td>AC1_2</td>
<td>28</td>
<td>100</td>
<td>40</td>
<td>79</td>
<td>89</td>
<td>64</td>
<td>101</td>
<td>83</td>
<td>101</td>
<td>100</td>
<td>97</td>
<td>89</td>
<td>89</td>
</tr>
<tr>
<td>AC1_3</td>
<td>161</td>
<td>119</td>
<td>182</td>
<td>184</td>
<td>148</td>
<td>115</td>
<td>118</td>
<td>169</td>
<td>128</td>
<td>143</td>
<td>112</td>
<td>116</td>
<td>206</td>
</tr>
<tr>
<td>AC1_4</td>
<td>191</td>
<td>209</td>
<td>208</td>
<td>205</td>
<td>197</td>
<td>197</td>
<td>136</td>
<td>208</td>
<td>148</td>
<td>187</td>
<td>197</td>
<td>172</td>
<td>209</td>
</tr>
<tr>
<td>TMCC_1</td>
<td>70</td>
<td>17</td>
<td>44</td>
<td>49</td>
<td>83</td>
<td>85</td>
<td>23</td>
<td>25</td>
<td>86</td>
<td>47</td>
<td>31</td>
<td>61</td>
<td>101</td>
</tr>
<tr>
<td>TMCC_2</td>
<td>133</td>
<td>194</td>
<td>155</td>
<td>139</td>
<td>169</td>
<td>209</td>
<td>178</td>
<td>125</td>
<td>152</td>
<td>157</td>
<td>191</td>
<td>193</td>
<td>131</td>
</tr>
<tr>
<td><strong>Mode 3</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AC1_1</td>
<td>10</td>
<td>20</td>
<td>4</td>
<td>98</td>
<td>11</td>
<td>76</td>
<td>7</td>
<td>61</td>
<td>35</td>
<td>8</td>
<td>53</td>
<td>74</td>
<td>40</td>
</tr>
<tr>
<td>AC1_2</td>
<td>28</td>
<td>40</td>
<td>89</td>
<td>101</td>
<td>101</td>
<td>97</td>
<td>89</td>
<td>100</td>
<td>79</td>
<td>64</td>
<td>83</td>
<td>100</td>
<td>89</td>
</tr>
<tr>
<td>AC1_3</td>
<td>161</td>
<td>182</td>
<td>148</td>
<td>118</td>
<td>128</td>
<td>112</td>
<td>206</td>
<td>197</td>
<td>148</td>
<td>197</td>
<td>209</td>
<td>209</td>
<td>205</td>
</tr>
<tr>
<td>AC1_4</td>
<td>191</td>
<td>208</td>
<td>197</td>
<td>136</td>
<td>148</td>
<td>197</td>
<td>209</td>
<td>209</td>
<td>205</td>
<td>197</td>
<td>208</td>
<td>187</td>
<td>172</td>
</tr>
<tr>
<td>AC1_5</td>
<td>277</td>
<td>251</td>
<td>224</td>
<td>269</td>
<td>290</td>
<td>256</td>
<td>226</td>
<td>236</td>
<td>220</td>
<td>314</td>
<td>227</td>
<td>292</td>
<td>223</td>
</tr>
<tr>
<td>AC1_6</td>
<td>316</td>
<td>295</td>
<td>280</td>
<td>299</td>
<td>316</td>
<td>305</td>
<td>244</td>
<td>256</td>
<td>305</td>
<td>317</td>
<td>317</td>
<td>313</td>
<td>305</td>
</tr>
<tr>
<td>AC1_7</td>
<td>335</td>
<td>400</td>
<td>331</td>
<td>385</td>
<td>359</td>
<td>332</td>
<td>377</td>
<td>398</td>
<td>364</td>
<td>334</td>
<td>344</td>
<td>328</td>
<td>422</td>
</tr>
<tr>
<td>AC1_8</td>
<td>425</td>
<td>421</td>
<td>413</td>
<td>424</td>
<td>403</td>
<td>388</td>
<td>407</td>
<td>424</td>
<td>413</td>
<td>352</td>
<td>364</td>
<td>413</td>
<td>425</td>
</tr>
<tr>
<td>TMCC_1</td>
<td>70</td>
<td>44</td>
<td>83</td>
<td>23</td>
<td>86</td>
<td>31</td>
<td>101</td>
<td>17</td>
<td>49</td>
<td>85</td>
<td>25</td>
<td>47</td>
<td>61</td>
</tr>
<tr>
<td>TMCC_2</td>
<td>131</td>
<td>155</td>
<td>169</td>
<td>178</td>
<td>152</td>
<td>191</td>
<td>131</td>
<td>194</td>
<td>139</td>
<td>209</td>
<td>125</td>
<td>157</td>
<td>193</td>
</tr>
<tr>
<td>TMCC_3</td>
<td>233</td>
<td>265</td>
<td>301</td>
<td>241</td>
<td>263</td>
<td>277</td>
<td>286</td>
<td>260</td>
<td>299</td>
<td>239</td>
<td>302</td>
<td>247</td>
<td>317</td>
</tr>
<tr>
<td>TMCC_4</td>
<td>410</td>
<td>355</td>
<td>425</td>
<td>341</td>
<td>373</td>
<td>409</td>
<td>349</td>
<td>371</td>
<td>385</td>
<td>394</td>
<td>368</td>
<td>407</td>
<td>347</td>
</tr>
</tbody>
</table>

### Table 4-10. Initial Sets of PRBS Register

<table>
<thead>
<tr>
<th>Segment Number</th>
<th>Initial Sets for Mode 1†</th>
<th>Initial Sets for Mode 2†</th>
<th>Initial Sets for Mode 3†</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>1 1 1 1 1 1 1 1 1 1</td>
<td>1 1 1 1 1 1 1 1 1 1</td>
<td>1 1 1 1 1 1 1 1 1 1</td>
</tr>
<tr>
<td>9</td>
<td>1 1 0 1 0 0 1 1 1 1</td>
<td>0 1 1 0 1 0 1 1 1 0</td>
<td>1 1 0 1 1 0 0 1 0 1</td>
</tr>
<tr>
<td>7</td>
<td>0 1 1 0 1 0 1 1 1 0</td>
<td>1 1 0 1 1 0 0 1 0 1</td>
<td>1 0 0 1 0 1 0 0 0 0</td>
</tr>
<tr>
<td>5</td>
<td>0 1 0 0 0 1 0 1 1 0</td>
<td>1 1 0 0 1 0 0 0 0 10</td>
<td>0 1 1 1 0 0 1 0 0 1</td>
</tr>
<tr>
<td>3</td>
<td>1 1 0 1 1 1 0 0 1 0 1</td>
<td>1 0 0 1 0 1 0 0 0 0</td>
<td>0 0 1 0 0 0 1 1 0 1</td>
</tr>
</tbody>
</table>
ISDB-T Components

Table 4-10. Initial Sets of PRBS Register (continued)

<table>
<thead>
<tr>
<th></th>
<th>00101111010</th>
<th>00010111000</th>
<th>11100110110</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>11001000010</td>
<td>01110001001</td>
<td>00100010111</td>
</tr>
<tr>
<td>2</td>
<td>00010000100</td>
<td>00000100100</td>
<td>11100111101</td>
</tr>
<tr>
<td>4</td>
<td>10010100000</td>
<td>00100011001</td>
<td>01101010011</td>
</tr>
<tr>
<td>6</td>
<td>11110110000</td>
<td>01100111001</td>
<td>10111010010</td>
</tr>
<tr>
<td>8</td>
<td>00001011000</td>
<td>11100110110</td>
<td>01100100010</td>
</tr>
<tr>
<td>10</td>
<td>10100100111</td>
<td>00101010001</td>
<td>11101001011</td>
</tr>
<tr>
<td>12</td>
<td>01110001001</td>
<td>00100001011</td>
<td>00010011100</td>
</tr>
</tbody>
</table>

†Degree from 0 to 10 in Figure 4-11

References

**DTV_MuxDiffSegs**

**Description**  
ISDB-T multiplex differential segments

**Library**  
DTV, ISDB-T

**Class**  
SDFDTV_MuxDiffSegs

**Parameters**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Default</th>
<th>Type</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Carriers</td>
<td>number of carriers in one segment</td>
<td>432</td>
<td>int</td>
<td>108 for mode 1; 216 for mode 2; 432 for mode 3</td>
</tr>
<tr>
<td>Segments</td>
<td>number of segments</td>
<td>1</td>
<td>int</td>
<td>[1, 13]</td>
</tr>
<tr>
<td>Start_Seg</td>
<td>initial number of segment (0 to Segments-1)</td>
<td>0</td>
<td>int</td>
<td>[0, 12]</td>
</tr>
</tbody>
</table>

**Pin Inputs**

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>data</td>
<td>TSP data input</td>
<td>complex</td>
</tr>
<tr>
<td>2</td>
<td>TMCC</td>
<td>TMCC data input</td>
<td>complex</td>
</tr>
<tr>
<td>3</td>
<td>AC1</td>
<td>AC1 data input</td>
<td>complex</td>
</tr>
<tr>
<td>4</td>
<td>AC2</td>
<td>AC2 data input</td>
<td>complex</td>
</tr>
</tbody>
</table>

**Pin Outputs**

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>output</td>
<td>differential segments data output</td>
<td>complex</td>
</tr>
</tbody>
</table>

**Notes/Equations**

1. The model is used to multiplex TSP (transport stream packet), TMCC (transmission and multiplexing configuration control), AC1 (auxiliary channel...
1), and AC2 (auxiliary channel 2) data into the differential modulation OFDM segments according to Figure 4-12 and Table 4-11.

2. IntState of Length determines the number of TMCC, AC1, and AC2 in each segment. IntState of Start_Seg and Segments determine the TMCC, AC1, and AC2 positions in each corresponding segment according to Table 4-11.

The PRBS sequences in each segment are generated according to Figure 4-13 and the initial sets of PRBS register in Table 4-12. The PRBS in initialized so that the first output bit from the PRBS coincides with the first active carrier. A new value is generated by the PRBS on every carrier used in each segment (whether or not it is a pilot).

After determining all TMCC, AC1, and AC2 positions in each corresponding segment and the value of the PRBS sequence in all the active carriers in each segment, TMCC, AC1, AC2, and TSP data are multiplexed into differential segments.

\[ x[\text{TMCC position}[i]] = \text{PilotValue}[\text{TMCC position}[i]] \times \text{TMCC} \]

where \( NTMCC \) is the number of TMCC in one segment; \( \text{PilotValue} \) is the PRBS sequence in the corresponding segment; \( x[i] \) is the output segment data.

The output of the input AC1 data is:

\[ x[\text{AC1 position}[i]] = \text{PilotValue}[\text{AC1 position}[i]] \times AC1[i] \]

AC2 data is:

\[ x[\text{AC2 position}[i]] = \text{PilotValue}[\text{AC2 position}[i]] \times AC2[i] \]

Except for the AC1, AC2, and TMCC data positions, the remaining positions in one segment are the TSP data positions.

\[ x[i] = \text{data}[i] \]
**Figure 4-12. Structure of OFDM Segment for Differential Modulation**

$S_{i,j,k}$ denotes complex data in the data segment after time and frequency interleaving.

$N_c=108$ for Mode 1; 216 for Mode 2; 432 for Mode 3

**Figure 4-13. Generation of PRBS Sequence**

$g(x) = x^{17} + x^{9} + 1$

$\text{Output} = W_i$
### Table 4-11. Carrier Allocation of CP, AC, and TMCC for Differential Modulation

<table>
<thead>
<tr>
<th>Segment</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Mode 1</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CP</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>AC1_1</td>
<td>10</td>
<td>53</td>
<td>61</td>
<td>11</td>
<td>20</td>
<td>74</td>
<td>35</td>
<td>76</td>
<td>4</td>
<td>40</td>
<td>8</td>
<td>798</td>
</tr>
<tr>
<td>AC1_2</td>
<td>28</td>
<td>83</td>
<td>100</td>
<td>101</td>
<td>40</td>
<td>100</td>
<td>79</td>
<td>97</td>
<td>89</td>
<td>89</td>
<td>64</td>
<td>89</td>
</tr>
<tr>
<td>AC1_3</td>
<td>3</td>
<td>3</td>
<td>29</td>
<td>28</td>
<td>23</td>
<td>30</td>
<td>3</td>
<td>5</td>
<td>13</td>
<td>72</td>
<td>36</td>
<td>25</td>
</tr>
<tr>
<td>AC2_1</td>
<td>45</td>
<td>15</td>
<td>41</td>
<td>45</td>
<td>63</td>
<td>81</td>
<td>72</td>
<td>18</td>
<td>93</td>
<td>95</td>
<td>48</td>
<td>30</td>
</tr>
<tr>
<td>AC2_2</td>
<td>59</td>
<td>40</td>
<td>84</td>
<td>81</td>
<td>85</td>
<td>92</td>
<td>85</td>
<td>57</td>
<td>98</td>
<td>100</td>
<td>52</td>
<td>42</td>
</tr>
<tr>
<td>AC2_3</td>
<td>77</td>
<td>58</td>
<td>93</td>
<td>91</td>
<td>105</td>
<td>103</td>
<td>89</td>
<td>92</td>
<td>102</td>
<td>105</td>
<td>74</td>
<td>104</td>
</tr>
<tr>
<td>AC2_4</td>
<td>13</td>
<td>25</td>
<td>4</td>
<td>36</td>
<td>10</td>
<td>7</td>
<td>49</td>
<td>31</td>
<td>16</td>
<td>5</td>
<td>78</td>
<td>34</td>
</tr>
<tr>
<td>AC2_5</td>
<td>50</td>
<td>63</td>
<td>7</td>
<td>48</td>
<td>28</td>
<td>25</td>
<td>61</td>
<td>39</td>
<td>30</td>
<td>10</td>
<td>82</td>
<td>48</td>
</tr>
<tr>
<td>AC2_6</td>
<td>70</td>
<td>73</td>
<td>17</td>
<td>55</td>
<td>44</td>
<td>47</td>
<td>96</td>
<td>47</td>
<td>37</td>
<td>21</td>
<td>85</td>
<td>54</td>
</tr>
<tr>
<td>TMCC_1</td>
<td>83</td>
<td>80</td>
<td>51</td>
<td>59</td>
<td>47</td>
<td>60</td>
<td>99</td>
<td>65</td>
<td>74</td>
<td>44</td>
<td>98</td>
<td>70</td>
</tr>
<tr>
<td>TMCC_2</td>
<td>87</td>
<td>93</td>
<td>71</td>
<td>86</td>
<td>54</td>
<td>87</td>
<td>104</td>
<td>72</td>
<td>83</td>
<td>61</td>
<td>102</td>
<td>101</td>
</tr>
<tr>
<td><strong>Mode 2</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CP</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>AC1_1</td>
<td>10</td>
<td>61</td>
<td>20</td>
<td>35</td>
<td>4</td>
<td>8</td>
<td>98</td>
<td>53</td>
<td>11</td>
<td>74</td>
<td>76</td>
<td>40</td>
</tr>
<tr>
<td>AC1_2</td>
<td>28</td>
<td>100</td>
<td>40</td>
<td>79</td>
<td>89</td>
<td>64</td>
<td>101</td>
<td>83</td>
<td>101</td>
<td>100</td>
<td>97</td>
<td>89</td>
</tr>
<tr>
<td>AC1_3</td>
<td>161</td>
<td>119</td>
<td>182</td>
<td>184</td>
<td>148</td>
<td>115</td>
<td>118</td>
<td>169</td>
<td>128</td>
<td>143</td>
<td>112</td>
<td>116</td>
</tr>
<tr>
<td>AC1_4</td>
<td>191</td>
<td>209</td>
<td>208</td>
<td>205</td>
<td>197</td>
<td>197</td>
<td>136</td>
<td>208</td>
<td>148</td>
<td>167</td>
<td>197</td>
<td>172</td>
</tr>
<tr>
<td>AC2_1</td>
<td>3</td>
<td>29</td>
<td>23</td>
<td>3</td>
<td>13</td>
<td>36</td>
<td>10</td>
<td>3</td>
<td>28</td>
<td>30</td>
<td>5</td>
<td>72</td>
</tr>
<tr>
<td>AC2_2</td>
<td>45</td>
<td>41</td>
<td>63</td>
<td>72</td>
<td>93</td>
<td>48</td>
<td>30</td>
<td>15</td>
<td>45</td>
<td>81</td>
<td>18</td>
<td>95</td>
</tr>
<tr>
<td>AC2_3</td>
<td>59</td>
<td>84</td>
<td>85</td>
<td>85</td>
<td>98</td>
<td>52</td>
<td>55</td>
<td>40</td>
<td>81</td>
<td>92</td>
<td>57</td>
<td>100</td>
</tr>
<tr>
<td>AC2_4</td>
<td>77</td>
<td>93</td>
<td>105</td>
<td>89</td>
<td>102</td>
<td>74</td>
<td>81</td>
<td>58</td>
<td>91</td>
<td>103</td>
<td>92</td>
<td>105</td>
</tr>
<tr>
<td>AC2_6</td>
<td>111</td>
<td>136</td>
<td>138</td>
<td>113</td>
<td>180</td>
<td>133</td>
<td>111</td>
<td>137</td>
<td>131</td>
<td>111</td>
<td>121</td>
<td>144</td>
</tr>
<tr>
<td>AC2_7</td>
<td>123</td>
<td>153</td>
<td>189</td>
<td>126</td>
<td>203</td>
<td>138</td>
<td>153</td>
<td>149</td>
<td>171</td>
<td>180</td>
<td>201</td>
<td>156</td>
</tr>
<tr>
<td>AC2_8</td>
<td>148</td>
<td>189</td>
<td>200</td>
<td>165</td>
<td>208</td>
<td>150</td>
<td>167</td>
<td>192</td>
<td>193</td>
<td>193</td>
<td>206</td>
<td>160</td>
</tr>
<tr>
<td>AC2_9</td>
<td>166</td>
<td>199</td>
<td>211</td>
<td>200</td>
<td>213</td>
<td>212</td>
<td>185</td>
<td>201</td>
<td>213</td>
<td>197</td>
<td>210</td>
<td>182</td>
</tr>
<tr>
<td>TMCC_1</td>
<td>13</td>
<td>4</td>
<td>10</td>
<td>49</td>
<td>16</td>
<td>78</td>
<td>23</td>
<td>25</td>
<td>36</td>
<td>7</td>
<td>31</td>
<td>5</td>
</tr>
<tr>
<td>TMCC_2</td>
<td>50</td>
<td>7</td>
<td>28</td>
<td>61</td>
<td>30</td>
<td>82</td>
<td>37</td>
<td>63</td>
<td>48</td>
<td>25</td>
<td>39</td>
<td>10</td>
</tr>
<tr>
<td>TMCC_3</td>
<td>70</td>
<td>17</td>
<td>44</td>
<td>96</td>
<td>37</td>
<td>85</td>
<td>51</td>
<td>73</td>
<td>55</td>
<td>47</td>
<td>47</td>
<td>51</td>
</tr>
<tr>
<td>TMCC_4</td>
<td>83</td>
<td>51</td>
<td>47</td>
<td>99</td>
<td>74</td>
<td>98</td>
<td>88</td>
<td>80</td>
<td>59</td>
<td>60</td>
<td>65</td>
<td>44</td>
</tr>
<tr>
<td>TMCC_5</td>
<td>87</td>
<td>71</td>
<td>54</td>
<td>104</td>
<td>83</td>
<td>102</td>
<td>105</td>
<td>93</td>
<td>86</td>
<td>87</td>
<td>72</td>
<td>61</td>
</tr>
</tbody>
</table>

4-52 DTV_MuxDiffSegs
Table 4-11. Carrier Allocation of CP, AC, and TMCC for Differential Modulation (continued)

<table>
<thead>
<tr>
<th>Segment</th>
<th>11</th>
<th>9</th>
<th>7</th>
<th>5</th>
<th>3</th>
<th>1</th>
<th>0</th>
<th>2</th>
<th>4</th>
<th>6</th>
<th>8</th>
<th>10</th>
<th>12</th>
</tr>
</thead>
<tbody>
<tr>
<td>TMCC_6</td>
<td>133</td>
<td>144</td>
<td>115</td>
<td>139</td>
<td>113</td>
<td>142</td>
<td>121</td>
<td>112</td>
<td>118</td>
<td>157</td>
<td>124</td>
<td>186</td>
<td>131</td>
</tr>
<tr>
<td>TMCC_7</td>
<td>171</td>
<td>156</td>
<td>133</td>
<td>147</td>
<td>118</td>
<td>156</td>
<td>158</td>
<td>115</td>
<td>136</td>
<td>169</td>
<td>138</td>
<td>190</td>
<td>145</td>
</tr>
<tr>
<td>TMCC_8</td>
<td>181</td>
<td>163</td>
<td>155</td>
<td>155</td>
<td>129</td>
<td>162</td>
<td>178</td>
<td>125</td>
<td>152</td>
<td>204</td>
<td>145</td>
<td>193</td>
<td>159</td>
</tr>
<tr>
<td>TMCC_9</td>
<td>188</td>
<td>167</td>
<td>168</td>
<td>173</td>
<td>152</td>
<td>178</td>
<td>191</td>
<td>159</td>
<td>155</td>
<td>207</td>
<td>182</td>
<td>206</td>
<td>176</td>
</tr>
<tr>
<td>TMCC_10</td>
<td>201</td>
<td>194</td>
<td>195</td>
<td>180</td>
<td>169</td>
<td>209</td>
<td>195</td>
<td>179</td>
<td>162</td>
<td>212</td>
<td>191</td>
<td>210</td>
<td>213</td>
</tr>
</tbody>
</table>

Mode 3

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>CP</td>
<td>10</td>
<td>20</td>
<td>4</td>
<td>98</td>
<td>11</td>
<td>76</td>
<td>7</td>
<td>61</td>
<td>35</td>
<td>8</td>
<td>53</td>
<td>74</td>
<td>40</td>
</tr>
<tr>
<td>AC1_1</td>
<td>28</td>
<td>40</td>
<td>89</td>
<td>101</td>
<td>101</td>
<td>97</td>
<td>89</td>
<td>100</td>
<td>79</td>
<td>64</td>
<td>83</td>
<td>100</td>
<td>89</td>
</tr>
<tr>
<td>AC1_3</td>
<td>161</td>
<td>182</td>
<td>148</td>
<td>118</td>
<td>128</td>
<td>112</td>
<td>206</td>
<td>119</td>
<td>184</td>
<td>115</td>
<td>169</td>
<td>143</td>
<td>116</td>
</tr>
<tr>
<td>AC1_4</td>
<td>191</td>
<td>208</td>
<td>197</td>
<td>136</td>
<td>148</td>
<td>197</td>
<td>209</td>
<td>209</td>
<td>205</td>
<td>197</td>
<td>208</td>
<td>187</td>
<td>172</td>
</tr>
<tr>
<td>AC1_5</td>
<td>277</td>
<td>251</td>
<td>224</td>
<td>269</td>
<td>290</td>
<td>256</td>
<td>226</td>
<td>236</td>
<td>220</td>
<td>314</td>
<td>227</td>
<td>292</td>
<td>223</td>
</tr>
<tr>
<td>AC1_6</td>
<td>316</td>
<td>295</td>
<td>280</td>
<td>299</td>
<td>316</td>
<td>305</td>
<td>244</td>
<td>256</td>
<td>305</td>
<td>317</td>
<td>317</td>
<td>313</td>
<td>305</td>
</tr>
<tr>
<td>AC1_7</td>
<td>335</td>
<td>400</td>
<td>331</td>
<td>385</td>
<td>359</td>
<td>332</td>
<td>377</td>
<td>398</td>
<td>364</td>
<td>334</td>
<td>344</td>
<td>328</td>
<td>422</td>
</tr>
<tr>
<td>AC1_8</td>
<td>425</td>
<td>421</td>
<td>413</td>
<td>424</td>
<td>403</td>
<td>388</td>
<td>407</td>
<td>424</td>
<td>413</td>
<td>352</td>
<td>364</td>
<td>413</td>
<td>425</td>
</tr>
<tr>
<td>AC2_1</td>
<td>3</td>
<td>23</td>
<td>13</td>
<td>10</td>
<td>28</td>
<td>5</td>
<td>25</td>
<td>29</td>
<td>3</td>
<td>36</td>
<td>3</td>
<td>30</td>
<td>72</td>
</tr>
<tr>
<td>AC2_2</td>
<td>45</td>
<td>63</td>
<td>93</td>
<td>30</td>
<td>45</td>
<td>18</td>
<td>30</td>
<td>41</td>
<td>72</td>
<td>48</td>
<td>15</td>
<td>81</td>
<td>95</td>
</tr>
<tr>
<td>AC2_3</td>
<td>59</td>
<td>85</td>
<td>98</td>
<td>55</td>
<td>81</td>
<td>57</td>
<td>42</td>
<td>84</td>
<td>85</td>
<td>52</td>
<td>40</td>
<td>92</td>
<td>100</td>
</tr>
<tr>
<td>AC2_4</td>
<td>77</td>
<td>105</td>
<td>102</td>
<td>81</td>
<td>91</td>
<td>92</td>
<td>104</td>
<td>93</td>
<td>89</td>
<td>74</td>
<td>58</td>
<td>103</td>
<td>105</td>
</tr>
<tr>
<td>AC2_6</td>
<td>111</td>
<td>138</td>
<td>180</td>
<td>111</td>
<td>131</td>
<td>121</td>
<td>118</td>
<td>136</td>
<td>113</td>
<td>133</td>
<td>137</td>
<td>111</td>
<td>144</td>
</tr>
<tr>
<td>AC2_7</td>
<td>123</td>
<td>189</td>
<td>203</td>
<td>153</td>
<td>171</td>
<td>201</td>
<td>138</td>
<td>153</td>
<td>126</td>
<td>138</td>
<td>149</td>
<td>180</td>
<td>156</td>
</tr>
<tr>
<td>AC2_8</td>
<td>148</td>
<td>200</td>
<td>208</td>
<td>167</td>
<td>193</td>
<td>206</td>
<td>163</td>
<td>189</td>
<td>165</td>
<td>150</td>
<td>192</td>
<td>193</td>
<td>160</td>
</tr>
<tr>
<td>AC2_9</td>
<td>166</td>
<td>211</td>
<td>213</td>
<td>185</td>
<td>213</td>
<td>210</td>
<td>189</td>
<td>199</td>
<td>200</td>
<td>212</td>
<td>201</td>
<td>197</td>
<td>182</td>
</tr>
<tr>
<td>AC2_10</td>
<td>216</td>
<td>216</td>
<td>216</td>
<td>216</td>
<td>216</td>
<td>216</td>
<td>216</td>
<td>216</td>
<td>216</td>
<td>216</td>
<td>216</td>
<td>216</td>
<td>216</td>
</tr>
<tr>
<td>AC2_11</td>
<td>245</td>
<td>219</td>
<td>252</td>
<td>219</td>
<td>246</td>
<td>288</td>
<td>219</td>
<td>239</td>
<td>229</td>
<td>226</td>
<td>244</td>
<td>221</td>
<td>241</td>
</tr>
<tr>
<td>AC2_12</td>
<td>257</td>
<td>288</td>
<td>264</td>
<td>231</td>
<td>297</td>
<td>311</td>
<td>261</td>
<td>279</td>
<td>309</td>
<td>246</td>
<td>261</td>
<td>234</td>
<td>246</td>
</tr>
<tr>
<td>AC2_13</td>
<td>300</td>
<td>301</td>
<td>268</td>
<td>256</td>
<td>308</td>
<td>316</td>
<td>275</td>
<td>301</td>
<td>314</td>
<td>271</td>
<td>297</td>
<td>273</td>
<td>258</td>
</tr>
<tr>
<td>AC2_14</td>
<td>309</td>
<td>305</td>
<td>290</td>
<td>274</td>
<td>319</td>
<td>321</td>
<td>293</td>
<td>321</td>
<td>318</td>
<td>297</td>
<td>307</td>
<td>308</td>
<td>320</td>
</tr>
<tr>
<td>AC2_15</td>
<td>324</td>
<td>324</td>
<td>324</td>
<td>324</td>
<td>324</td>
<td>324</td>
<td>324</td>
<td>324</td>
<td>324</td>
<td>324</td>
<td>324</td>
<td>324</td>
<td>324</td>
</tr>
<tr>
<td>AC2_16</td>
<td>352</td>
<td>329</td>
<td>349</td>
<td>353</td>
<td>327</td>
<td>360</td>
<td>327</td>
<td>354</td>
<td>396</td>
<td>327</td>
<td>347</td>
<td>337</td>
<td>334</td>
</tr>
<tr>
<td>AC2_17</td>
<td>369</td>
<td>342</td>
<td>354</td>
<td>365</td>
<td>396</td>
<td>327</td>
<td>339</td>
<td>405</td>
<td>419</td>
<td>369</td>
<td>387</td>
<td>417</td>
<td>354</td>
</tr>
<tr>
<td>AC2_18</td>
<td>405</td>
<td>381</td>
<td>366</td>
<td>408</td>
<td>409</td>
<td>376</td>
<td>364</td>
<td>416</td>
<td>424</td>
<td>383</td>
<td>409</td>
<td>422</td>
<td>379</td>
</tr>
<tr>
<td>AC2_19</td>
<td>415</td>
<td>416</td>
<td>428</td>
<td>417</td>
<td>413</td>
<td>398</td>
<td>382</td>
<td>427</td>
<td>429</td>
<td>401</td>
<td>429</td>
<td>426</td>
<td>405</td>
</tr>
</tbody>
</table>
ISDB-T Components

Table 4-11. Carrier Allocation of CP, AC, and TMCC for Differential Modulation (continued)

<table>
<thead>
<tr>
<th>Segment</th>
<th>11</th>
<th>9</th>
<th>7</th>
<th>5</th>
<th>3</th>
<th>1</th>
<th>0</th>
<th>2</th>
<th>4</th>
<th>6</th>
<th>8</th>
<th>10</th>
<th>12</th>
</tr>
</thead>
<tbody>
<tr>
<td>TMCC_1</td>
<td>13</td>
<td>10</td>
<td>16</td>
<td>23</td>
<td>36</td>
<td>31</td>
<td>34</td>
<td>4</td>
<td>49</td>
<td>78</td>
<td>25</td>
<td>7</td>
<td>5</td>
</tr>
<tr>
<td>TMCC_2</td>
<td>50</td>
<td>28</td>
<td>30</td>
<td>37</td>
<td>48</td>
<td>39</td>
<td>48</td>
<td>7</td>
<td>61</td>
<td>82</td>
<td>63</td>
<td>25</td>
<td>10</td>
</tr>
<tr>
<td>TMCC_3</td>
<td>70</td>
<td>44</td>
<td>37</td>
<td>51</td>
<td>55</td>
<td>47</td>
<td>54</td>
<td>17</td>
<td>96</td>
<td>85</td>
<td>73</td>
<td>47</td>
<td>21</td>
</tr>
<tr>
<td>TMCC_4</td>
<td>83</td>
<td>47</td>
<td>74</td>
<td>68</td>
<td>59</td>
<td>65</td>
<td>70</td>
<td>51</td>
<td>99</td>
<td>98</td>
<td>80</td>
<td>60</td>
<td>44</td>
</tr>
<tr>
<td>TMCC_5</td>
<td>87</td>
<td>54</td>
<td>83</td>
<td>105</td>
<td>86</td>
<td>72</td>
<td>101</td>
<td>71</td>
<td>104</td>
<td>102</td>
<td>93</td>
<td>87</td>
<td>61</td>
</tr>
<tr>
<td>TMCC_6</td>
<td>133</td>
<td>115</td>
<td>113</td>
<td>121</td>
<td>118</td>
<td>124</td>
<td>131</td>
<td>144</td>
<td>139</td>
<td>142</td>
<td>112</td>
<td>157</td>
<td>186</td>
</tr>
<tr>
<td>TMCC_7</td>
<td>171</td>
<td>133</td>
<td>118</td>
<td>158</td>
<td>136</td>
<td>138</td>
<td>145</td>
<td>156</td>
<td>147</td>
<td>156</td>
<td>115</td>
<td>169</td>
<td>190</td>
</tr>
<tr>
<td>TMCC_8</td>
<td>181</td>
<td>155</td>
<td>129</td>
<td>178</td>
<td>152</td>
<td>145</td>
<td>159</td>
<td>163</td>
<td>155</td>
<td>162</td>
<td>125</td>
<td>204</td>
<td>193</td>
</tr>
<tr>
<td>TMCC_9</td>
<td>188</td>
<td>168</td>
<td>152</td>
<td>191</td>
<td>155</td>
<td>182</td>
<td>176</td>
<td>167</td>
<td>173</td>
<td>178</td>
<td>159</td>
<td>207</td>
<td>206</td>
</tr>
<tr>
<td>TMCC_10</td>
<td>201</td>
<td>195</td>
<td>169</td>
<td>195</td>
<td>162</td>
<td>191</td>
<td>213</td>
<td>194</td>
<td>180</td>
<td>209</td>
<td>179</td>
<td>212</td>
<td>210</td>
</tr>
<tr>
<td>TMCC_11</td>
<td>220</td>
<td>265</td>
<td>294</td>
<td>241</td>
<td>223</td>
<td>221</td>
<td>229</td>
<td>226</td>
<td>232</td>
<td>239</td>
<td>252</td>
<td>247</td>
<td>250</td>
</tr>
<tr>
<td>TMCC_12</td>
<td>223</td>
<td>277</td>
<td>298</td>
<td>279</td>
<td>241</td>
<td>222</td>
<td>229</td>
<td>226</td>
<td>246</td>
<td>253</td>
<td>264</td>
<td>255</td>
<td>264</td>
</tr>
<tr>
<td>TMCC_13</td>
<td>233</td>
<td>312</td>
<td>301</td>
<td>289</td>
<td>263</td>
<td>237</td>
<td>286</td>
<td>260</td>
<td>253</td>
<td>267</td>
<td>271</td>
<td>263</td>
<td>270</td>
</tr>
<tr>
<td>TMCC_14</td>
<td>267</td>
<td>315</td>
<td>314</td>
<td>296</td>
<td>276</td>
<td>260</td>
<td>299</td>
<td>263</td>
<td>290</td>
<td>284</td>
<td>275</td>
<td>281</td>
<td>286</td>
</tr>
<tr>
<td>TMCC_15</td>
<td>287</td>
<td>320</td>
<td>318</td>
<td>309</td>
<td>303</td>
<td>277</td>
<td>303</td>
<td>270</td>
<td>299</td>
<td>321</td>
<td>302</td>
<td>288</td>
<td>317</td>
</tr>
<tr>
<td>TMCC_16</td>
<td>360</td>
<td>355</td>
<td>358</td>
<td>328</td>
<td>373</td>
<td>402</td>
<td>349</td>
<td>331</td>
<td>329</td>
<td>337</td>
<td>334</td>
<td>340</td>
<td>347</td>
</tr>
<tr>
<td>TMCC_17</td>
<td>372</td>
<td>363</td>
<td>372</td>
<td>331</td>
<td>385</td>
<td>406</td>
<td>387</td>
<td>349</td>
<td>334</td>
<td>374</td>
<td>352</td>
<td>354</td>
<td>361</td>
</tr>
<tr>
<td>TMCC_18</td>
<td>379</td>
<td>371</td>
<td>376</td>
<td>341</td>
<td>420</td>
<td>409</td>
<td>397</td>
<td>371</td>
<td>345</td>
<td>394</td>
<td>368</td>
<td>361</td>
<td>375</td>
</tr>
<tr>
<td>TMCC_19</td>
<td>383</td>
<td>389</td>
<td>394</td>
<td>375</td>
<td>423</td>
<td>422</td>
<td>404</td>
<td>384</td>
<td>368</td>
<td>407</td>
<td>371</td>
<td>398</td>
<td>392</td>
</tr>
<tr>
<td>TMCC_20</td>
<td>410</td>
<td>396</td>
<td>425</td>
<td>395</td>
<td>428</td>
<td>426</td>
<td>417</td>
<td>411</td>
<td>385</td>
<td>411</td>
<td>378</td>
<td>407</td>
<td>429</td>
</tr>
</tbody>
</table>

Table 4-12. Initial Sets of PRBS Register

<table>
<thead>
<tr>
<th>Segment</th>
<th>Initial Sets for Mode 1</th>
<th>Initial Sets for Mode 2</th>
<th>Initial Sets for Mode 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>1 1 1 1 1 1 1 1 1 1</td>
<td>1 1 1 1 1 1 1 1 1 1</td>
<td>1 1 1 1 1 1 1 1 1 1</td>
</tr>
<tr>
<td>9</td>
<td>1 1 0 1 1 0 0 1 1 1</td>
<td>0 1 1 0 1 0 1 1 1 1</td>
<td>1 1 0 1 1 0 0 1 1 0</td>
</tr>
<tr>
<td>7</td>
<td>0 1 1 0 1 0 1 1 1 0</td>
<td>1 1 0 1 1 0 0 1 0 1</td>
<td>1 0 0 1 0 0 0 0 0 0</td>
</tr>
<tr>
<td>5</td>
<td>0 1 0 0 0 1 0 1 1 0</td>
<td>1 1 0 0 1 0 0 0 1 0</td>
<td>0 1 1 1 0 0 1 0 0 1</td>
</tr>
<tr>
<td>3</td>
<td>1 1 0 1 1 0 0 1 0 1</td>
<td>1 0 0 1 0 1 0 0 0 0</td>
<td>0 0 1 0 0 1 1 0 0 1</td>
</tr>
<tr>
<td>1</td>
<td>0 0 1 0 1 1 1 0 1 0</td>
<td>0 0 0 0 1 0 1 0 0 0</td>
<td>1 1 0 0 1 1 1 0 1 0</td>
</tr>
<tr>
<td>0</td>
<td>1 1 0 0 1 0 0 0 0 1</td>
<td>0 1 1 0 0 0 1 0 0 1</td>
<td>0 0 1 0 0 0 1 0 1 1</td>
</tr>
<tr>
<td>2</td>
<td>0 0 0 1 0 0 0 0 1 0</td>
<td>0 0 0 0 0 1 0 0 1 0</td>
<td>1 1 0 0 1 1 1 0 1 0</td>
</tr>
<tr>
<td>4</td>
<td>1 0 0 1 0 1 0 0 0 0</td>
<td>0 0 1 0 0 1 1 0 0 1</td>
<td>0 1 1 0 1 0 0 1 1 1</td>
</tr>
<tr>
<td>6</td>
<td>1 1 1 0 1 1 0 0 0 0</td>
<td>0 1 1 0 0 1 1 0 0 1</td>
<td>1 0 1 1 0 1 0 0 1 0</td>
</tr>
</tbody>
</table>
Table 4-12. Initial Sets of PRBS Register

<table>
<thead>
<tr>
<th>Segment</th>
<th>Initial Sets for Mode 1†</th>
<th>Initial Sets for Mode 2†</th>
<th>Initial Sets for Mode 3†</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>0 0 0 0 1 0 1 1 0 0 0</td>
<td>1 1 1 0 0 1 1 0 1 1 0</td>
<td>0 1 1 0 0 0 1 0 0 1 0</td>
</tr>
<tr>
<td>10</td>
<td>1 0 1 0 0 1 0 0 1 1 1</td>
<td>0 0 1 0 1 0 1 0 0 0 1</td>
<td>1 1 1 0 1 0 0 0 1 0 1</td>
</tr>
<tr>
<td>12</td>
<td>0 1 1 1 0 0 0 1 0 0 1</td>
<td>0 0 1 0 0 0 1 0 1 1 1</td>
<td>0 0 0 1 0 0 1 1 1 0 0</td>
</tr>
</tbody>
</table>

†Degree from 0 to 10 in Figure 4-13.

References

ISDB-T Components

**DTV_PackTMCC**

**Description**  Complete TMCC bits (204 bits)

**Library**  DTV, ISDB-T

**Class**  SDFDTV_PackTMCC

**Parameters**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Default</th>
<th>Type</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Length</td>
<td>length of TMCC transmission format</td>
<td>204</td>
<td>int</td>
<td>204</td>
</tr>
<tr>
<td>InformLength</td>
<td>length of TMCC information after CDSC</td>
<td>184</td>
<td>int</td>
<td>184</td>
</tr>
<tr>
<td>InitBit</td>
<td>initialization bit for the DBPSK modulation (1 bit)</td>
<td>1</td>
<td>int</td>
<td>{0, 1}</td>
</tr>
<tr>
<td>SynchWord</td>
<td>synchronization word (16 bits): W0, W1</td>
<td>W0</td>
<td>enum</td>
<td></td>
</tr>
<tr>
<td>SegDesc</td>
<td>segment descriptor (3 bits): Differential, Coherent</td>
<td>Differential</td>
<td>enum</td>
<td></td>
</tr>
</tbody>
</table>

**Pin Inputs**

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>input</td>
<td>TMCC information bits after CDSC coding</td>
<td>int</td>
</tr>
</tbody>
</table>

**Pin Outputs**

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>output</td>
<td>complete TMCC transmission format</td>
<td>int</td>
</tr>
</tbody>
</table>

**Notes/Equations**

1. The model is used to multiplex the CDSC coded TMCC information bits (184 bits) and other no-coded 20 bits, which include initialization bit, synchronization of TMCC and system type.
According to ISDB-T, the first bit $B_0$ is the initialization bit for DBPSK modulation; $B_0$ to $B_{16}$ is a 16-bit synchronization sequence that takes $w_0$ and $w_1$ (the inverse of $w_0$) in turn in every frame; $B_{17}$ to $B_{19}$ represent the segment descriptor (differential modulation 111, coherent modulation 000); $B_{20}$ to $B_{203}$ include TMCC 102 information bits and 82-bit parity (these 184 bits are CDSC coded).

References

ISDB-T Components

**DTV_TMCCDemod**

**Description**
TMCC differential demodulation

**Library**
DTV, ISDB-T

**Class**
SDFDTV_TMCCDemod

**Parameters**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Default</th>
<th>Type</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Length</td>
<td>length of TMCC bits per OFDM frame</td>
<td>204</td>
<td>int</td>
<td>204</td>
</tr>
</tbody>
</table>

**Pin Inputs**

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>input</td>
<td>TMCC format (204 bits) before demodulation in the receiver</td>
<td>complex</td>
</tr>
</tbody>
</table>

**Pin Outputs**

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>output</td>
<td>demodulated TMCC transmission format</td>
<td>int</td>
</tr>
</tbody>
</table>

**Notes/Equations**

1. The model is used to perform DBPSK demodulation for 204 complex signals about the received TMCC format signal.

   Based on the received signal, this model makes hard decision on the real part of the complex signal.
where $0 \leq i < 204$. Then set $B_0 = B'\{0\}$, the DBPSK demodulation as follows:

$$B_i = B[i] \oplus B'[i-1]$$

where $i = 1, 2, ..., 203$.

References

ISDB-T Components

**DTV_TMCCInfo**

**Description** TMCC information for 102 bits from b20 to b121 in TMCC bit assignment

**Library** DTV, ISDB-T

**Class** SDFDTV_TMCCInfo

**Parameters**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Default</th>
<th>Type</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Length</td>
<td>length of TMCC information bits</td>
<td>102</td>
<td>int</td>
<td>[102]</td>
</tr>
<tr>
<td>Description</td>
<td>system description (2 bits): ISDB-T, Reserved Des 1, Reserved Des 2, Reserved Des 3</td>
<td>ISDB-T</td>
<td>enum</td>
<td></td>
</tr>
<tr>
<td>Count</td>
<td>index for transmission parameter change (4 bits): Ordinary, Frames 15, Frames 14, Frames 13, Frames 12, Frames 11, Frames 10, Frames 9, Frames 8, Frames 7, Frames 6, Frames 5, Frames 4, Frames 3, Frames 2, Frames 1</td>
<td>Ordinary</td>
<td>enum</td>
<td></td>
</tr>
<tr>
<td>Flag</td>
<td>control flag for alert broadcasting (1 bit): Ordinary Control, Switch-on</td>
<td>Ordinary Control</td>
<td>enum</td>
<td></td>
</tr>
<tr>
<td>CurFlag</td>
<td>current partial reception layer (1 bit): Unused Cur Flag, Used Cur Flag</td>
<td>Used Cur Flag</td>
<td>enum</td>
<td></td>
</tr>
<tr>
<td>CurA_Mod</td>
<td>current modulation for Layer A (3 bits): CurA DQPSK, CurA QPSK, CurA 16QAM, CurA 64QAM, CurA Reserved Mod 1, CurA Reserved Mod 2, CurA Reserved Mod 3, CurA Unused Layer Mod</td>
<td>CurA DQPSK</td>
<td>enum</td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
<td>Default</td>
<td>Type</td>
<td>Range</td>
</tr>
<tr>
<td>--------------</td>
<td>------------------------------------------------------------------------------</td>
<td>---------</td>
<td>--------</td>
<td>------------------------------------</td>
</tr>
<tr>
<td>CurA_Interlv</td>
<td>current time interleaver for Layer A (3 bits): CurA Int 0 0 0, CurA Int 4 2 1, CurA Int 8 4 2, CurA Int 16 8 4, CurA Int 32 16 8, CurA Reserved Int 1, CurA Reserved Int 2, CurA Unused Layer Int</td>
<td>CurA Int 8 4 2</td>
<td>enum</td>
<td></td>
</tr>
<tr>
<td>CurA_NumSeg</td>
<td>current number of segments for Layer A (4 bits): CurA Reserved Seg 1, CurA Seg 1, CurA Seg 2, CurA Seg 3, CurA Seg 4, CurA Seg 5, CurA Seg 6, CurA Seg 7, CurA Seg 8, CurA Seg 9, CurA Seg 10, CurA Seg 11, CurA Seg 12, CurA Seg 13, CurA Reserved Seg 2, CurA Unused Layer Seg</td>
<td>CurA Seg 13</td>
<td>enum</td>
<td></td>
</tr>
<tr>
<td>CurB_Interlv</td>
<td>current time interleaver for Layer B (3 bits): CurB Int 0 0 0, CurB Int 4 2 1, CurB Int 8 4 2, CurB Int 16 8 4, CurB Int 32 16 8, CurB Reserved Int 1, CurB Reserved Int 2, CurB Unused Layer Int</td>
<td>CurB Int 8 4 2</td>
<td>enum</td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
<td>Default</td>
<td>Type</td>
<td>Range</td>
</tr>
<tr>
<td>--------------</td>
<td>------------------------------------------------------------------------------</td>
<td>------------------------------------------</td>
<td>--------</td>
<td>---------------</td>
</tr>
<tr>
<td>CurB_NumSeg</td>
<td>current number of segments for Layer B (4 bits): CurB Reserved Seg 1, CurB Seg 1, CurB Seg 2, CurB Seg 3, CurB Seg 4, CurB Seg 5, CurB Seg 6, CurB Seg 7, CurB Seg 8, CurB Seg 9, CurB Seg 10, CurB Seg 11, CurB Seg 12, CurB Seg 13, CurB Reserved Seg 2, CurB Unused Layer Seg</td>
<td>CurB Unused Layer Seg</td>
<td>enum</td>
<td></td>
</tr>
<tr>
<td>CurC_Mod</td>
<td>current modulation for Layer C (3 bits): CurC DQPSK, CurC QPSK, CurC 16QAM, CurC 64QAM, CurC Reserved Mod 1, CurC Reserved Mod 2, CurC Reserved Mod 3, CurC Unused Layer Mod</td>
<td>CurC 64QAM</td>
<td>enum</td>
<td></td>
</tr>
<tr>
<td>CurC_Rate</td>
<td>current code rate for Layer C (3 bits): CurC 1/2, CurC 2/3, CurC 3/4, CurC 5/6, CurC 7/8, CurC Reserved Cod 1, CurC Reserved Cod 2, CurC Unused Layer Cod</td>
<td>CurC 1/2</td>
<td>enum</td>
<td></td>
</tr>
<tr>
<td>CurC_Interlv</td>
<td>current time interleaver for Layer C (3 bits): CurC Int 0 0 0, CurC Int 4 2 1, CurC Int 8 4 2, CurC Int 16 8 4, CurC Int 32 16 8, CurC Reserved Int 1, CurC Reserved Int 2, CurC Unused Layer Int</td>
<td>CurC Int 8 4 2</td>
<td>enum</td>
<td></td>
</tr>
<tr>
<td>CurC_NumSeg</td>
<td>current number of segments for Layer C (4 bits): CurC Reserved Seg 1, CurC Seg 1, CurC Seg 2, CurC Seg 3, CurC Seg 4, CurC Seg 5, CurC Seg 6, CurC Seg 7, CurC Seg 8, CurC Seg 9, CurC Seg 10, CurC Seg 11, CurC Seg 12, CurC Seg 13, CurC Reserved Seg 2, CurC Unused Layer Seg</td>
<td>CurC Seg 6</td>
<td>enum</td>
<td></td>
</tr>
<tr>
<td>NextFlag</td>
<td>next partial reception layer (1 bit): Unused Next Flag, Used Next</td>
<td>Unused Next Flag</td>
<td>enum</td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
<td>Default</td>
<td>Type</td>
<td>Range</td>
</tr>
<tr>
<td>--------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>----------------------------------</td>
<td>--------</td>
<td>-------</td>
</tr>
<tr>
<td>NextA_Mod</td>
<td>next modulation for Layer A (3 bits): NextA DQPSK, NextA QPSK, NextA 16QAM, NextA 64QAM, NextA Reserved Mod 1, NextA Reserved Mod 2, NextA Reserved Mod 3, NextA Unused Layer Mod</td>
<td>NextA QPSK</td>
<td>enum</td>
<td></td>
</tr>
<tr>
<td>NextA_Rate</td>
<td>next code rate for Layer A (3 bits): NextA 1/2, NextA 2/3, NextA 3/4, NextA 5/6, NextA 7/8, NextA Reserved Cod 1, NextA Reserved Cod 2, NextA Reserved Cod 3, NextA Unused Layer Cod</td>
<td>NextA 1/2</td>
<td>enum</td>
<td></td>
</tr>
<tr>
<td>NextA_Interlv</td>
<td>next time interleaving for Layer A (3 bits): NextA Int 000, NextA Int 4 2 1, NextA Int 8 4 2, NextA Int 16 8 4, NextA Int 32 16 8, NextA Reserved Int 1, NextA Reserved Int 2, NextA Unused Layer Int</td>
<td>NextA Int 8 4 2</td>
<td>enum</td>
<td></td>
</tr>
<tr>
<td>NextA_NumSeg</td>
<td>next number of segments for Layer A (4 bits): NextA Reserved Seg 1, NextA Seg 1, NextA Seg 2, NextA Seg 3, NextA Seg 4, NextA Seg 5, NextA Seg 6, NextA Seg 7, NextA Seg 8, NextA Seg 9, NextA Seg 10, NextA Seg 11, NextA Seg 12, NextA Seg 13, NextA Reserved Seg 2, NextA Unused Layer Seg</td>
<td>NextA Reserved Seg 1</td>
<td>enum</td>
<td></td>
</tr>
<tr>
<td>NextB_Mod</td>
<td>next modulation for Layer B (3 bits): NextB DQPSK, NextB QPSK, NextB 16QAM, NextB 64QAM, NextB Reserved Mod 1, NextB Reserved Mod 2, NextB Reserved Mod 3, NextB Unused Layer Mod</td>
<td>NextB 16QAM</td>
<td>enum</td>
<td></td>
</tr>
<tr>
<td>NextB_Rate</td>
<td>next code rate for Layer B (3 bits): NextB 1/2, NextB 2/3, NextB 3/4, NextB 5/6, NextB 7/8, NextB Reserved Cod 1, NextB Reserved Cod 2, NextB Unused Layer Cod</td>
<td>NextB 1/2</td>
<td>enum</td>
<td></td>
</tr>
</tbody>
</table>
### ISDB-T Components

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Default</th>
<th>Type</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>NextB_Interlv</td>
<td>next time interleaving for Layer B (3 bits): NextB Int 0 0 0, NextB Int 4 2 1, NextB Int 8 4 2, NextB Int 16 8 4, NextB Int 32 16 8, NextB Reserved Int 1, NextB Reserved Int 2, NextB Unused Layer Int</td>
<td>NextB Int 8 4 2</td>
<td>enum</td>
<td></td>
</tr>
<tr>
<td>NextB_NumSeg</td>
<td>next number of segments for Layer B (4 bits): NextB Reserved Seg 1, NextB Seg 1, NextB Seg 2, NextB Seg 3, NextB Seg 4, NextB Seg 5, NextB Seg 6, NextB Seg 7, NextB Seg 8, NextB Seg 9, NextB Seg 10, NextB Seg 11, NextB Seg 12, NextB Seg 13, NextB Reserved Seg 2, NextB Unused Layer Seg</td>
<td>NextB Seg 6</td>
<td>enum</td>
<td></td>
</tr>
<tr>
<td>NextC_Mod</td>
<td>next modulation for Layer C (3 bits): NextC DQPSK, NextC QPSK, NextC 16QAM, NextC 64QAM, NextC Reserved Mod 1, NextC Reserved Mod 2, NextC Reserved Mod 3, NextC Unused Layer Mod</td>
<td>NextC QPSK</td>
<td>enum</td>
<td></td>
</tr>
<tr>
<td>NextC_Rate</td>
<td>next code rate for Layer C (3 bits): NextC 1/2, NextC 2/3, NextC 3/4, NextC 5/6, NextC 7/8, NextC Reserved Cod 1, NextC Reserved Cod 2, NextC Unused Layer Cod</td>
<td>NextC 1/2</td>
<td>enum</td>
<td></td>
</tr>
<tr>
<td>NextC_Interlv</td>
<td>next time interleaving for Layer C (3 bits): NextC Int 0 0 0, NextC Int 4 2 1, NextC Int 8 4 2, NextC Int 16 8 4, NextC Int 32 16 8, NextC Reserved Int 1, NextC Reserved Int 2, NextC Unused Layer Int</td>
<td>NextC Int 8 4 2</td>
<td>enum</td>
<td></td>
</tr>
</tbody>
</table>
1. This model is used to generate the 102 bits of TMCC information according to Table 4-13.

2. For the parameters in this model, TMCC information bits are generated according to Table 4-13; specification bits assignments are given in Table 4-14 through Table 4-22.

### Pin Outputs

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>output</td>
<td>TMCC information bits (B20-B121)</td>
<td>int</td>
</tr>
</tbody>
</table>

### Notes/Equations

1. This model is used to generate the 102 bits of TMCC information according to Table 4-13.

2. For the parameters in this model, TMCC information bits are generated according to Table 4-13; specification bits assignments are given in Table 4-14 through Table 4-22.

### Table 4-13. Bit Assignments for TMCC Information

<table>
<thead>
<tr>
<th>Bits</th>
<th>No. of Bits</th>
<th>Purpose/Content</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>$B_{20} - B_{21}$</td>
<td>2</td>
<td>System descriptor</td>
<td>Table 4-15</td>
</tr>
<tr>
<td>$B_{22} - B_{25}$</td>
<td>4</td>
<td>Count-down index</td>
<td>Table 4-16</td>
</tr>
<tr>
<td>$B_{26}$</td>
<td>1</td>
<td>Switch-on control flag used for alert broadcasting</td>
<td>Table 4-17</td>
</tr>
<tr>
<td>$B_{27}$</td>
<td>1</td>
<td>Current configuration information</td>
<td>Table 4-18</td>
</tr>
<tr>
<td>$B_{28} - B_{40}$</td>
<td>13</td>
<td>Transmission parameters for layer a</td>
<td>Table 4-14</td>
</tr>
<tr>
<td>$B_{41} - B_{53}$</td>
<td>13</td>
<td>Transmission parameters for layer b</td>
<td></td>
</tr>
<tr>
<td>$B_{54} - B_{66}$</td>
<td>13</td>
<td>Transmission parameters for layer c</td>
<td></td>
</tr>
</tbody>
</table>
### Table 4-13. Bit Assignments for TMCC Information

<table>
<thead>
<tr>
<th>Bits</th>
<th>No. of Bits</th>
<th>Purpose/Content</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>$B_{67}$</td>
<td>1</td>
<td>Next configuration information</td>
<td></td>
</tr>
<tr>
<td>$B_{68} - B_{80}$</td>
<td>13</td>
<td>Transmission parameters for layer a</td>
<td>Table 4-14</td>
</tr>
<tr>
<td>$B_{81} - B_{93}$</td>
<td>13</td>
<td>Transmission parameters for layer b</td>
<td>Table 4-14</td>
</tr>
<tr>
<td>$B_{94} - B_{106}$</td>
<td>13</td>
<td>Transmission parameters for layer c</td>
<td>Table 4-14</td>
</tr>
<tr>
<td>$B_{107} - B_{121}$</td>
<td>15</td>
<td>(reserved for future use)</td>
<td>All set to “1”</td>
</tr>
</tbody>
</table>

### Table 4-14. Transmission Parameters

<table>
<thead>
<tr>
<th></th>
<th>No. of Bits</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Modulation</td>
<td>3</td>
<td>Table 4-19</td>
</tr>
<tr>
<td>Code Rate</td>
<td>3</td>
<td>Table 4-20</td>
</tr>
<tr>
<td>Time Interleaving</td>
<td>3</td>
<td>Table 4-21</td>
</tr>
<tr>
<td>Number of Segments</td>
<td>4</td>
<td>Table 4-22</td>
</tr>
</tbody>
</table>

### Table 4-15. System Descriptor

<table>
<thead>
<tr>
<th>$B_{20} - B_{21}$</th>
<th>System</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>ISDB-T using 13 segments</td>
</tr>
<tr>
<td>01,10,11</td>
<td>(reserved for future use)</td>
</tr>
</tbody>
</table>

### Table 4-16. Count-Down Index

<table>
<thead>
<tr>
<th>$B_{22} - B_{25}$</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1111</td>
<td>15 Frames before changing transmission parameters</td>
</tr>
<tr>
<td>1110</td>
<td>14 Frames before changing transmission parameters</td>
</tr>
<tr>
<td>1101</td>
<td>13 Frames before changing transmission parameters</td>
</tr>
<tr>
<td>1100</td>
<td>12 Frames before changing transmission parameters</td>
</tr>
<tr>
<td></td>
<td>.</td>
</tr>
</tbody>
</table>
### Table 4-16. Count-Down Index

<table>
<thead>
<tr>
<th>B_{22} – B_{25}</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0010</td>
<td>3 Frames before changing transmission parameters</td>
</tr>
<tr>
<td>0001</td>
<td>2 Frames before changing transmission parameters</td>
</tr>
<tr>
<td>0000</td>
<td>1 Frames before changing transmission parameters</td>
</tr>
</tbody>
</table>

### Table 4-17. Switch-on Control Flag used for Albert Broadcasting

<table>
<thead>
<tr>
<th>B_{26}</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Ordinary</td>
</tr>
<tr>
<td>1</td>
<td>Switch-on</td>
</tr>
</tbody>
</table>

### Table 4-18. Partial Reception Flag

<table>
<thead>
<tr>
<th>B_{26}, B_{27}</th>
<th>Partial Reception Layer</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Not used</td>
</tr>
<tr>
<td>1</td>
<td>Used</td>
</tr>
</tbody>
</table>

### Table 4-19. Modulation Scheme of OFDM Carrier

<table>
<thead>
<tr>
<th>Modulation</th>
</tr>
</thead>
<tbody>
<tr>
<td>000</td>
</tr>
<tr>
<td>001</td>
</tr>
<tr>
<td>010</td>
</tr>
<tr>
<td>011</td>
</tr>
<tr>
<td>100-110</td>
</tr>
<tr>
<td>111</td>
</tr>
</tbody>
</table>

### Table 4-20. Code Rate of Inner Code

<table>
<thead>
<tr>
<th>Code Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>000</td>
</tr>
<tr>
<td>001</td>
</tr>
<tr>
<td>010</td>
</tr>
</tbody>
</table>
ISDB-T Components

**Table 4-20. Code Rate of Inner Code**

<table>
<thead>
<tr>
<th>Code Rate</th>
<th>011</th>
<th>5/6</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>100</td>
<td>7/8</td>
</tr>
<tr>
<td></td>
<td>101-110</td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td>111</td>
<td>Not used</td>
</tr>
</tbody>
</table>

**Table 4-21. Time Interleaving**

<table>
<thead>
<tr>
<th>Time Interleaving Parameter I</th>
<th>001</th>
<th>4(Mode 1), 4(Mode 2), 4(Mode 3)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>010</td>
<td>8(Mode 1), 8(Mode 2), 8(Mode 3)</td>
</tr>
<tr>
<td></td>
<td>011</td>
<td>16(Mode 1), 16(Mode 2), 16(Mode 3)</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>32(Mode 1), 32(Mode 2), 32(Mode 3)</td>
</tr>
<tr>
<td></td>
<td>101-110</td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td>111</td>
<td>Not used</td>
</tr>
</tbody>
</table>

**Table 4-22. Number of Segments**

<table>
<thead>
<tr>
<th>No. of Segments used in the Layer</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000</td>
</tr>
<tr>
<td>0001</td>
</tr>
<tr>
<td>0010</td>
</tr>
<tr>
<td>0011</td>
</tr>
<tr>
<td>0100</td>
</tr>
<tr>
<td>0101</td>
</tr>
<tr>
<td>0110</td>
</tr>
<tr>
<td>0111</td>
</tr>
<tr>
<td>1000</td>
</tr>
<tr>
<td>1001</td>
</tr>
<tr>
<td>1010</td>
</tr>
<tr>
<td>1011</td>
</tr>
<tr>
<td>1100</td>
</tr>
<tr>
<td>1101</td>
</tr>
</tbody>
</table>
Table 4-22. Number of Segments

<table>
<thead>
<tr>
<th>No. of Segments used in the Layer</th>
</tr>
</thead>
<tbody>
<tr>
<td>1110</td>
</tr>
<tr>
<td>1111</td>
</tr>
</tbody>
</table>

References

ISDB-T Components

**DTV_TMCCMod**

**Description**  
TMCC differential modulation

**Library**  
DTV, ISDB-T

**Class**  
SDFDTV_TMCCMod

**Parameters**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Default</th>
<th>Type</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Length</td>
<td>length of TMCC bits per OFDM frame</td>
<td>204</td>
<td>int</td>
<td>204</td>
</tr>
</tbody>
</table>

**Pin Inputs**

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>input</td>
<td>received TMCC transmission format (204 bits) before modulation in the transmitter</td>
<td>int</td>
</tr>
</tbody>
</table>

**Pin Outputs**

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>output</td>
<td>modulated TMCC transmission format</td>
<td>complex</td>
</tr>
</tbody>
</table>

**Notes/Equations**

1. The model is used to perform DBPSK modulation.
2. $B'[0] = B_0$; DBPSK modulation is:
   \[
   B'[i] = B_i \oplus B'_{i-1}
   \]
   where $i = 1, 2, ..., 203$.

   Then
coded bits $B'[0] \sim B'[203]$ are converted to $\{1.0, 0\}, \{1.0, 0\}$.
where $i=0,1, \ldots, 203$.

References

ISDB-T Components

**DTV_TimeInterlv**

**Description**  Interleaver and deinterleaver of complex data  
**Library**  DTV, ISDB-T  
**Class**  SDFDTV_TimeInterlv

**Parameters**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Default</th>
<th>Type</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Carriers</td>
<td>numbers of carriers in each segment for specific OFDM modulation mode</td>
<td>96</td>
<td>int</td>
<td>(96, 192, 384)†</td>
</tr>
<tr>
<td>Segments</td>
<td>number of segments to be interleaved simultaneously</td>
<td>13</td>
<td>int</td>
<td>[1, 13]</td>
</tr>
<tr>
<td>Option</td>
<td>option for interleaving or de-interleaving: Interleave, Deinterleave</td>
<td>Interleave</td>
<td>enum</td>
<td></td>
</tr>
<tr>
<td>Initial_value</td>
<td>initial value in interleaver delay FIFOs</td>
<td>0.0+j*0.0</td>
<td>complex</td>
<td>[0, ∞)</td>
</tr>
<tr>
<td>I</td>
<td>factor to multiply when calculating delay period for interleaver branches</td>
<td>0</td>
<td>int</td>
<td>[0, 32]</td>
</tr>
</tbody>
</table>

† Carriers = 96, 192, 384 for mode 1, 2, 3, respectively

**Pin Inputs**

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>input</td>
<td>input symbols to be interleaved</td>
<td>complex</td>
</tr>
</tbody>
</table>

**Pin Outputs**

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>output</td>
<td>output symbols after interleaved</td>
<td>complex</td>
</tr>
</tbody>
</table>
1. This model is used to perform symbol-wise interleaving over the output symbols of constellation mapping and modulation.

2. Time interleaving performed for each segment data is carried out through $n_c$ delay branches, where $n_c$ is the number of carriers per segment. The delay of each branch is:

$$D = I \times m_i \quad m_i = \{I \times 5\} \mod n_c$$

where $I$ is a parameter for each segment described in Table 4-23.

<table>
<thead>
<tr>
<th>Mode</th>
<th>I</th>
<th>No. of Symbols for Delay Adjustment</th>
<th>Number of OFDM Frames to be delayed by Delay Adjustment and Time Interleaving</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mode 1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>28</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>56</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>16</td>
<td>112</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>224</td>
<td>16</td>
</tr>
<tr>
<td>Mode 2</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>14</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>28</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>16</td>
<td>56</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>112</td>
<td>8</td>
</tr>
<tr>
<td>Mode 3</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>109</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>14</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>16</td>
<td>28</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>56</td>
<td>4</td>
</tr>
</tbody>
</table>

References

ISDB-T Components
ISDB-T Components
ISDB-T Components
Chapter 5: Multiplex Components
Multiplex Components

DTV_CommCtrl2

**Description**  2-input commutator with input particle number control
**Library**  DTV, Multiplex
**Class**  SDFDTV_CommCtrl2

**Parameters**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Default</th>
<th>Type</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>NumInput1</td>
<td>number of particles from input 1</td>
<td>1</td>
<td>int</td>
<td>[1, ∞)</td>
</tr>
<tr>
<td>NumInput2</td>
<td>number of particles from input 2</td>
<td>1</td>
<td>int</td>
<td>[1, ∞)</td>
</tr>
</tbody>
</table>

**Pin Inputs**

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>in1</td>
<td>input 1</td>
<td>anytype</td>
</tr>
<tr>
<td>2</td>
<td>in2</td>
<td>input 2</td>
<td>anytype</td>
</tr>
</tbody>
</table>

**Pin Outputs**

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>output</td>
<td>output comprised of two inputs</td>
<td>anytype</td>
</tr>
</tbody>
</table>

**Notes/Equations**

1. This model is used to combine two input signals into one. NumInput1 of input 1 and NumOutput2 of input 2 data particles are combined and output.
DTV_CommCtrl3

Description  3-input commutator with input particle number control
Library  DTV, Multiplex
Class  SDFDTV_CommCtrl3

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Default</th>
<th>Type</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>NumInput1</td>
<td>number of particles from input 1</td>
<td>1</td>
<td>int</td>
<td>[1, ∞)</td>
</tr>
<tr>
<td>NumInput2</td>
<td>number of particles from input 2</td>
<td>1</td>
<td>int</td>
<td>[1, ∞)</td>
</tr>
<tr>
<td>NumInput3</td>
<td>number of particles from input 3</td>
<td>1</td>
<td>int</td>
<td>[1, ∞)</td>
</tr>
</tbody>
</table>

Pin Inputs

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>in1</td>
<td>input 1</td>
<td>anytype</td>
</tr>
<tr>
<td>2</td>
<td>in2</td>
<td>input 2</td>
<td>anytype</td>
</tr>
<tr>
<td>3</td>
<td>in3</td>
<td>input 3</td>
<td>anytype</td>
</tr>
</tbody>
</table>

Pin Outputs

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>output</td>
<td>output comprised of three inputs</td>
<td>anytype</td>
</tr>
</tbody>
</table>

Notes/Equations

1. This model is used to combine three input signals into one.

   NumInput1 of input 1, NumOutput2 of input 2, and NumInput3 of input 3 data particles are combined and output.
Multiplex Components

DTV_DistCtrl2

Description  2-output distributor with output particle number control
Library  DTV, Multiplex
Class  SDFDTV_DistCtrl2

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Default</th>
<th>Type</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>NumOutput1</td>
<td>number of particles directed to output 1</td>
<td>1</td>
<td>int</td>
<td>[1, ∞)</td>
</tr>
<tr>
<td>NumOutput2</td>
<td>number of particles directed to output 2</td>
<td>1</td>
<td>int</td>
<td>[1, ∞)</td>
</tr>
</tbody>
</table>

Pin Inputs

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>input</td>
<td>input to be distributed over the two outputs</td>
<td>anytype</td>
</tr>
</tbody>
</table>

Pin Outputs

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>out1</td>
<td>output 1</td>
<td>anytype</td>
</tr>
<tr>
<td>3</td>
<td>out2</td>
<td>output 2</td>
<td>anytype</td>
</tr>
</tbody>
</table>

Notes/Equations

1. This model is used to distribute one data stream to two outputs. NumOutput1 and NumOutput2 data particles are distributed to output 1, and output 2, respectively.
**DTV_DistCtrl3**

Description  3-output distributor with output particle number control  
Library  DTV, Multiplex  
Class  SDFDTV_DistCtrl3

**Parameters**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Default</th>
<th>Type</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>NumOutput1</td>
<td>number of particles directed to output 1</td>
<td>1</td>
<td>int</td>
<td>[1, ∞)</td>
</tr>
<tr>
<td>NumOutput2</td>
<td>number of particles directed to output 2</td>
<td>1</td>
<td>int</td>
<td>[1, ∞)</td>
</tr>
<tr>
<td>NumOutput3</td>
<td>number of particles directed to output 3</td>
<td>1</td>
<td>int</td>
<td>[1, ∞)</td>
</tr>
</tbody>
</table>

**Pin Inputs**

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>input</td>
<td>input to be distributed over the three outputs</td>
<td>anytype</td>
</tr>
</tbody>
</table>

**Pin Outputs**

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>out1</td>
<td>output 1</td>
<td>anytype</td>
</tr>
<tr>
<td>3</td>
<td>out2</td>
<td>output 2</td>
<td>anytype</td>
</tr>
<tr>
<td>4</td>
<td>out3</td>
<td>output 3</td>
<td>anytype</td>
</tr>
</tbody>
</table>

**Notes/Equations**

1. This model is used to distribute one data stream to three outputs.  
   NumOutput1, NumOutput2 and NumOutput3 data particles are distributed to output 1, output 2, and output 3, respectively.
Multiplex Components

**DTV_SplitThreeLayData**

**Description**  Data stream splitter into 3-layer data

**Library**  DTV, Multiplex

**Class**  SDFDTV_SplitThreeLayData

**Parameters**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Default</th>
<th>Type</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Carriers</td>
<td>number of carriers in one segment</td>
<td>432</td>
<td>int</td>
<td>[96, 108, 192, 216, 384, 432]†</td>
</tr>
<tr>
<td>SegmentsA</td>
<td>number of segments in layer A</td>
<td>1</td>
<td>int</td>
<td>[1]††</td>
</tr>
<tr>
<td>SegmentsB</td>
<td>number of segments in layer B</td>
<td>1</td>
<td>int</td>
<td>[1, 12]††</td>
</tr>
<tr>
<td>SegmentsC</td>
<td>number of segments in layer C</td>
<td>1</td>
<td>int</td>
<td>[1, 12]††</td>
</tr>
</tbody>
</table>

† According to ISDB-T, the Carriers are 96 and 108, 192 and 216, 384 and 432, corresponding to mode 1, mode 2, and mode 3, respectively.

†† The sum of SegmentsA, SegmentsB, and SegmentsC ≤ 13.

**Pin Inputs**

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Synt</td>
<td>synthesis of 3-layer data streams</td>
<td>complex</td>
</tr>
</tbody>
</table>

**Pin Outputs**

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>LayA</td>
<td>layer A output</td>
<td>complex</td>
</tr>
<tr>
<td>3</td>
<td>LayB</td>
<td>layer B output</td>
<td>complex</td>
</tr>
<tr>
<td>4</td>
<td>LayC</td>
<td>layer C output</td>
<td>complex</td>
</tr>
</tbody>
</table>
Notes/Equations

1. The model is used to demultiplex one synthetic stream into a 3-layer data stream, for use in ISDB-T 3-layer systems only.

Each firing, Synt consumes \((\text{Segments}_A + \text{Segments}_B + \text{Segments}_C) \times \text{Carriers tokens}\); LayA produces \(\text{Segments}_A \times \text{Carriers tokens}\); LayB produces \(\text{Segments}_B \times \text{Carriers tokens}\); LayC produces \(\text{Segments}_C \times \text{Carriers tokens}\).

References

**Multiplex Components**

**DTV_SplitThreeLayTSP**

**Description**
TSP stream splitter into 3-layer TSP stream

**Library**
DTV, Multiplex

**Class**
SDFDTV_SplitThreeLayTSP

**Parameters**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Default</th>
<th>Type</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Length</td>
<td>number of bytes in one TSP</td>
<td>204</td>
<td>int</td>
<td>(204)</td>
</tr>
<tr>
<td>SegmentsA</td>
<td>number of segments in layer A</td>
<td>1</td>
<td>int</td>
<td>[1]†</td>
</tr>
<tr>
<td>LayerA_Modulation</td>
<td>modulation mode for layer A: A DQPSK, A QPSK, A 16QAM, A 64QAM</td>
<td>A DQPSK</td>
<td>enum</td>
<td></td>
</tr>
<tr>
<td>SegmentsB</td>
<td>number of segments in layer B</td>
<td>5</td>
<td>int</td>
<td>[1, 12]†</td>
</tr>
<tr>
<td>LayerB_Modulation</td>
<td>modulation mode for Layer B: B DQPSK, B QPSK, B 16QAM, B 64QAM</td>
<td>B QPSK</td>
<td>enum</td>
<td></td>
</tr>
<tr>
<td>LayerB_Convolutional_Code</td>
<td>convolutional code rate for layer B: B 1/2, B 2/3, B 3/4, B 5/6, B 7/8</td>
<td>B 7/8</td>
<td>enum</td>
<td></td>
</tr>
<tr>
<td>SegmentsC</td>
<td>number of segments in layer C</td>
<td>7</td>
<td>int</td>
<td>[1, 12]†</td>
</tr>
<tr>
<td>LayerC_Modulation</td>
<td>modulation mode for layer C: C DQPSK, C QPSK, C 16QAM, C 64QAM</td>
<td>C 64QAM</td>
<td>enum</td>
<td></td>
</tr>
<tr>
<td>LayerC_Convolutional_Code</td>
<td>convolutional code rate for layer C: C 1/2, C 2/3, C 3/4, C 5/6, C 7/8</td>
<td>C 7/8</td>
<td>enum</td>
<td></td>
</tr>
</tbody>
</table>

† The sum of SegmentsA, SegmentsB and SegmentsC ≤ 13.
Pin Inputs

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>input</td>
<td>Input TSP data stream to be split</td>
<td>int</td>
</tr>
</tbody>
</table>

Pin Outputs

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>OutA</td>
<td>layer A output</td>
<td>int</td>
</tr>
<tr>
<td>3</td>
<td>OutB</td>
<td>layer B output</td>
<td>int</td>
</tr>
<tr>
<td>4</td>
<td>OutC</td>
<td>layer C output</td>
<td>int</td>
</tr>
</tbody>
</table>

Notes/Equations

1. This model is used in ISDB-T 3-layer systems to split one TSP data stream into a 3-layer TSP data stream according to Table 5-1.

2. One TSP data stream is split into a 3-layer TSP data stream: layer A, layer B, and layer C.

   The number of TSPs per segment in each layer is determined according to modulation mode and convolutional code rate.

   The model determines the total number of TSPs in each layer by the number of TSPs per segments and the number of segments in each layer in one OFDM frame. To increase simulation speed, the total number of TSPs in each layer is divided by the greatest common divisor, which is the greatest common divisor of the total number of TSPs in each layer. An example follows for mode 1.

   SegmentsA=1
   LayerA_Modulation=A 64QAM
   LayerA_Convolutional_Code=A 2/3

   SegmentsB=5
   LayerB_Modulation=B DQPSK
   LayerB_Convolutional_Code=B 1/2

   SegmentsC=7
   LayerC_Modulation=C 16QAM
   LayerC_Convolutional_Code=C 3/4

   Referring to Table 5-1, the number of TSPs is 48, 12, and 36 per segment, respectively.
The total number of TSPs in each layer is $48 \times 1$, $12 \times 5$ and $36 \times 7$ in one OFDM frame; the greatest common divisor is 12. After dividing, the total number of TSPs is 4, 5 and 21, respectively. Then, the model splits $204 \times 4 \times 1$ integers to OutA pin, $204 \times 1 \times 5$ integers to OutB pin, and $204 \times 3 \times 7$ integers to OutC pin.

Table 5-1. Transmitting TSPs per Segment for ISDB-T

<table>
<thead>
<tr>
<th>Carrier Modulation</th>
<th>Convolutional Code</th>
<th>Number of Transmitting TSPs† (Mode 1/2/3)</th>
</tr>
</thead>
<tbody>
<tr>
<td>DQPSK, QPSK</td>
<td>1/2</td>
<td>12/24/48</td>
</tr>
<tr>
<td></td>
<td>2/3</td>
<td>16/32/64</td>
</tr>
<tr>
<td></td>
<td>3/4</td>
<td>18/36/72</td>
</tr>
<tr>
<td></td>
<td>5/6</td>
<td>20/40/80</td>
</tr>
<tr>
<td></td>
<td>7/8</td>
<td>21/42/84</td>
</tr>
<tr>
<td>16-QAM</td>
<td>1/2</td>
<td>24/48/96</td>
</tr>
<tr>
<td></td>
<td>2/3</td>
<td>32/64/128</td>
</tr>
<tr>
<td></td>
<td>3/4</td>
<td>36/72/144</td>
</tr>
<tr>
<td></td>
<td>5/6</td>
<td>40/80/160</td>
</tr>
<tr>
<td></td>
<td>7/8</td>
<td>42/84/168</td>
</tr>
<tr>
<td>64-QAM</td>
<td>1/2</td>
<td>36/72/144</td>
</tr>
<tr>
<td></td>
<td>2/3</td>
<td>48/96/192</td>
</tr>
<tr>
<td></td>
<td>3/4</td>
<td>54/108/216</td>
</tr>
<tr>
<td></td>
<td>5/6</td>
<td>60/120/240</td>
</tr>
<tr>
<td></td>
<td>7/8</td>
<td>63/126/252</td>
</tr>
</tbody>
</table>

† number of transmitting TSPs per one OFDM frame

References

**DTV_SplitTwoLayData**

**Description**  Data stream splitter into 2-layer data

**Library**  DTV, Multiplex

**Class**  SDFDTV_SplitTwoLayData

**Parameters**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Default</th>
<th>Type</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Carriers</td>
<td>number of carriers in one segment</td>
<td>432</td>
<td>int</td>
<td>[96, 108, 192, 216, 384, 432]†</td>
</tr>
<tr>
<td>SegmentsA</td>
<td>number of segments in layer A</td>
<td>1</td>
<td>int</td>
<td>[1, 12]††</td>
</tr>
<tr>
<td>SegmentsB</td>
<td>number of segments in layer B</td>
<td>1</td>
<td>int</td>
<td>[1, 12]††</td>
</tr>
</tbody>
</table>

† According to ISDB-T, the number of Carriers is 96 and 108, 192 and 216, 384 and 432, corresponding to mode 1, mode 2, and mode 3, respectively.

†† The sum of SegmentsA and SegmentsB ≤ 13.

**Pin Inputs**

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Synt</td>
<td>synthesis of 2-layer data stream</td>
<td>complex</td>
</tr>
</tbody>
</table>

**Pin Outputs**

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>LayA</td>
<td>layer A output</td>
<td>complex</td>
</tr>
<tr>
<td>3</td>
<td>LayB</td>
<td>layer B output</td>
<td>complex</td>
</tr>
</tbody>
</table>

**Notes/Equations**
Multiplex Components

1. The model is used in ISDB-T 2-layer systems to demultiplex one synthetic stream into a 2-layer data stream.

2. Each firing, Synt consumes \((\text{SegmentsA} + \text{SegmentsB}) \times \text{Carriers}\) tokens; LayA produces \(\text{SegmentsA} \times \text{Carriers}\) tokens; LayB produces \(\text{SegmentsB} \times \text{Carriers}\) tokens.

References

DTV_SplitTwoLayTSP

Description  TSP stream splitter into 2-layer TSP stream
Library     DTV, Multiplex
Class       SDFDTV_SplitTwoLayTSP

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Default</th>
<th>Type</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Length</td>
<td>number of bytes in one TSP</td>
<td>204</td>
<td>int</td>
<td>(204)</td>
</tr>
<tr>
<td>SegmentsA</td>
<td>number of segments in layer A</td>
<td>5</td>
<td>int</td>
<td>[1, 12]†</td>
</tr>
<tr>
<td>LayerA_Modulation</td>
<td>modulation mode for layer A: A DQPSK, A QPSK, A 16QAM, A 64QAM</td>
<td>A DQPSK</td>
<td>enum</td>
<td></td>
</tr>
<tr>
<td>SegmentsB</td>
<td>number of segments in layer B</td>
<td>8</td>
<td>int</td>
<td>[1, 12]†</td>
</tr>
<tr>
<td>LayerB_Modulation</td>
<td>modulation mode for layer B: B DQPSK, B QPSK, B 16QAM, B 64QAM</td>
<td>B DQPSK</td>
<td>enum</td>
<td></td>
</tr>
<tr>
<td>LayerB_Convolutional_Code</td>
<td>convolutional code rate for layer B: B 1/2, B 2/3, B 3/4, B 5/6, B 7/8</td>
<td>B 7/8</td>
<td>enum</td>
<td></td>
</tr>
</tbody>
</table>

† The sum of SegmentsA and SegmentsB ≤ 13.

Pin Inputs

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>input</td>
<td>Input TSP data stream to be split</td>
<td>int</td>
</tr>
</tbody>
</table>
Multiplex Components

Pin Outputs

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>OutA</td>
<td>layer A output</td>
<td>int</td>
</tr>
<tr>
<td>3</td>
<td>OutB</td>
<td>layer B output</td>
<td>int</td>
</tr>
</tbody>
</table>

Notes/Equations

1. This model is used in ISDB-T 2-layer systems to split one TSP data stream into a 2-layer TSP data stream according to Table 5-2.

2. The TSP stream is split into a 2-layer TSP data stream: layer A and layer B.
   The number of TSPs per segment in each layer is determined according to modulation mode and convolutional code rate.

   The model determines the total number of TSPs in each layer by the number of TSPs per segments and the number of segments in each layer in one OFDM frame. To increase simulation speed, the total number of TSPs in each layer is divided by the greatest common divisor, which is the greatest common divisor between the total number of TSPs in each layer. An example follows for mode 1.

   SegmentsA = 5
   LayerA_Modulation = A DQPSK
   LayerA_Convolutional_Code = A 1/2

   SegmentsB = 7
   LayerB_Modulation = B 16QAM
   LayerB_Convolutional_Code = B 3/4

   Referring to Table 5-2, the number of TSPs is 12 and 36 per segment, respectively. The total number of TSPs in each layer is 12×5 and 36×7 in one OFDM frame; the greatest common divisor is 12. After dividing, the total number of TSPs is 5 and 21, respectively. So, the model distributes 204×1×5 integers to OutA pin, and 204×3×7 integers to OutB pin.

Table 5-2. Transmitting TSPs per Segment for ISDB-T

<table>
<thead>
<tr>
<th>Carrier Modulation</th>
<th>Convolutional Code</th>
<th>Number of Transmitting TSPs T (Mode 1/2/3)</th>
</tr>
</thead>
</table>

5-14  DTV_SplitTwoLayerTSP
Table 5-2. Transmitting TSPs per Segment for ISDB-T

<table>
<thead>
<tr>
<th>DQPSK, QPSK</th>
<th>1/2</th>
<th>12/24/48</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2/3</td>
<td>16/32/64</td>
</tr>
<tr>
<td></td>
<td>3/4</td>
<td>18/36/72</td>
</tr>
<tr>
<td></td>
<td>5/6</td>
<td>20/40/80</td>
</tr>
<tr>
<td></td>
<td>7/8</td>
<td>21/42/84</td>
</tr>
<tr>
<td>16-QAM</td>
<td>1/2</td>
<td>24/48/96</td>
</tr>
<tr>
<td></td>
<td>2/3</td>
<td>32/64/128</td>
</tr>
<tr>
<td></td>
<td>3/4</td>
<td>36/72/144</td>
</tr>
<tr>
<td></td>
<td>5/6</td>
<td>40/80/160</td>
</tr>
<tr>
<td></td>
<td>7/8</td>
<td>42/84/168</td>
</tr>
<tr>
<td>64-QAM</td>
<td>1/2</td>
<td>36/72/144</td>
</tr>
<tr>
<td></td>
<td>2/3</td>
<td>48/96/192</td>
</tr>
<tr>
<td></td>
<td>3/4</td>
<td>54/108/216</td>
</tr>
<tr>
<td></td>
<td>5/6</td>
<td>60/120/240</td>
</tr>
<tr>
<td></td>
<td>7/8</td>
<td>63/126/252</td>
</tr>
</tbody>
</table>

† number of transmitting TSPs per one OFDM frame

References

**Multiplex Components**

**DTV_SynLayTMCC1**

**Description**  Synthesizer for 1-layer TMCC received into one TMCC format

**Library**  DTV, Multiplex

**Class**  SDFDTV_SynLayTMCC1

**Parameters**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Default</th>
<th>Sym</th>
<th>Type</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Segments</td>
<td>number of segments</td>
<td>1</td>
<td>seg</td>
<td>int</td>
<td>[1, 13]</td>
</tr>
</tbody>
</table>

**Pin Inputs**

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>input</td>
<td>1-layer input</td>
<td>complex</td>
</tr>
</tbody>
</table>

**Pin Outputs**

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>output</td>
<td>synthesis of 1-layer TMCC stream</td>
<td>complex</td>
</tr>
</tbody>
</table>

**Notes/Equations**

1. The model is used in ISDB-T 1-layer systems to synthesize one layer TMCC streams into one synthetic stream.

2. Because there is one received TMCC signal from each segment in every layer in the DTV_DemuxCohSegs and DTV_DemuxDiffSegs models, all TMCC received must be synthesized to one TMCC signal, as follows.

```c
Complex in(0.0,0.0), out(0.0,0.0);
for (int i=0; i<seg; i++)
in += (input%(i)).operator Complex();
out = in/seg;
output%(0) << out;
```
References

Multiplex Components

**DTV_SynLayTMCC2**

**Description**  Synthesizer for 2-layer TMCC received into one TMCC format  
**Library**  DTV, Multiplex  
**Class**  SDFDTV_SynLayTMCC2

### Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Default</th>
<th>Sym</th>
<th>Type</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>SegmentsA</td>
<td>number of segments in layer A</td>
<td>1</td>
<td>segA</td>
<td>int</td>
<td>[1, 12]†</td>
</tr>
<tr>
<td>SegmentsB</td>
<td>number of segments in layer B</td>
<td>1</td>
<td>segB</td>
<td>int</td>
<td>[1, 12]†</td>
</tr>
</tbody>
</table>

† The sum of SegmentsA and SegmentsB ≤ 13.

### Pin Inputs

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>LayA</td>
<td>layer A input</td>
<td>complex</td>
</tr>
<tr>
<td>2</td>
<td>LayB</td>
<td>layer B input</td>
<td>complex</td>
</tr>
</tbody>
</table>

### Pin Outputs

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>Synt</td>
<td>synthesis of 2-layer TMCC streams</td>
<td>complex</td>
</tr>
</tbody>
</table>

### Notes/Equations

1. The model is used to synthesize 2-layer TMCC streams into one synthetic stream, for use in ISDB-T 2-layer systems.
2. Because there is one received TMCC signal from each segment in every layer in the DTV_DemuxCohSegs and DTV_DemuxDiffSegs models, all received TMCC must be synthesized to one TMCC signal, as follows.

```cpp
Complex in1(0.0,0.0),in2(0.0,0.0),out(0.0,0.0);
for (int i=0; i<segA ;i++)
in1 += (LayA%(i)).operator Complex();
for ( i=0; i<segB ;i++)
in2 += (LayB%(i)).operator Complex();
out = (in1+in2)/sumAB;
Synt%(0) << out;
```

References

Multiplex Components

**DTV_SynLayTMCC3**

**Description** Synthesizer for 3-layer TMCC received into one TMCC format

**Library** DTV, Multiplex

**Class** SDFDTV_SynLayTMCC3

### Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Default</th>
<th>Sym</th>
<th>Type</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>SegmentsA</td>
<td>number of segments in layer A</td>
<td>1</td>
<td>segA</td>
<td>int</td>
<td>{1}† ††</td>
</tr>
<tr>
<td>SegmentsB</td>
<td>number of segments in layer B</td>
<td>1</td>
<td>segB</td>
<td>int</td>
<td>{1, 12}††</td>
</tr>
<tr>
<td>SegmentsC</td>
<td>number of segments in layer C</td>
<td>1</td>
<td>segC</td>
<td>int</td>
<td>{1, 12}††</td>
</tr>
</tbody>
</table>

† According to ISDB-T, the number of segments in the first layer must be 1.
†† The sum of SegmentsA, SegmentsB and SegmentsC ≤ 13.

### Pin Inputs

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>LayA</td>
<td>layer A input</td>
<td>complex</td>
</tr>
<tr>
<td>2</td>
<td>LayB</td>
<td>layer B input</td>
<td>complex</td>
</tr>
<tr>
<td>3</td>
<td>LayC</td>
<td>layer C input</td>
<td>complex</td>
</tr>
</tbody>
</table>

### Pin Outputs

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>Synt</td>
<td>synthesis of 3-layer TMCC streams</td>
<td>complex</td>
</tr>
</tbody>
</table>

### Notes/Equations

5-20 DTV_SynLayTMCC3
1. The model is used in ISDB-T 3-layer systems to synthesize 3-layer TMCC streams into one synthetic stream.

2. Because there is one received TMCC signal from each segment in every layer in the DTV_DemuxCohSegs and DTV_DemuxDiffSegs models, all received TMCC must be synthesized into one TMCC signal, as follows.

```c
Complex in1(0,0,0,0), in2(0,0,0,0), in3(0,0,0,0), out(0,0,0,0);
for (int i=0; i<segA; i++)
    in1 += (LayA%(i)).operator Complex();
for (i=0; i<segB; i++)
    in2 += (LayB%(i)).operator Complex();
for (i=0; i<segC; i++)
    in3 += (LayC%(i)).operator Complex();
out = (in1 + in2 + in3)/sumABC;
Synt%(0) <<= out;
```

References

Multiplex Components

**DTV_SynThreeLayData**

**Description**  Synthesizer for 3-layer data into one stream

**Library**  DTV, Multiplex

**Class**  SDFDTV_SynThreeLayData

### Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Default</th>
<th>Type</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Carriers</td>
<td>number of carriers in one segment</td>
<td>432</td>
<td>int</td>
<td>[96, 108, 192, 216, 384, 432]†</td>
</tr>
<tr>
<td>SegmentsA</td>
<td>number of segments in layer A</td>
<td>1</td>
<td>int</td>
<td>(1)†† †††</td>
</tr>
<tr>
<td>SegmentsB</td>
<td>number of segments in layer B</td>
<td>1</td>
<td>int</td>
<td>[1, 12]†††</td>
</tr>
<tr>
<td>SegmentsC</td>
<td>number of segments in layer C</td>
<td>1</td>
<td>int</td>
<td>[1, 12]†††</td>
</tr>
</tbody>
</table>

† According to ISDB-T, the Carriers should be 96 and 108, 192 and 216, 384 and 432, corresponding to mode 1, mode 2, and mode 3, respectively.
†† According to ISDB-T, the number of segments in the first layer must be 1.
††† The sum of SegmentsA, SegmentsB and SegmentsC ≤ 13.

### Pin Inputs

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>LayA</td>
<td>layer A input</td>
<td>complex</td>
</tr>
<tr>
<td>2</td>
<td>LayB</td>
<td>layer B input</td>
<td>complex</td>
</tr>
<tr>
<td>3</td>
<td>LayC</td>
<td>layer C input</td>
<td>complex</td>
</tr>
</tbody>
</table>

### Pin Outputs

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>Synt</td>
<td>synthesis of 3-layer data streams</td>
<td>complex</td>
</tr>
</tbody>
</table>
Notes/Equations

1. The model is used in ISDB-T 3-layer systems to multiplex 3-layer data streams into one synthetic stream.

2. Each firing, LayA consumes \(\text{Segments}A \times \text{Carriers}\) tokens; LayB consumes \(\text{Segments}B \times \text{Carriers}\) tokens; LayC consumes \(\text{Segments}C \times \text{Carriers}\) tokens; Synt produces \((\text{Segments}A + \text{Segments}B + \text{Segments}C) \times \text{Carriers}\) tokens.

References

Multiplex Components

**DTV_SynThreeLayTSP**

**Description**  Synthesizer for 3-layer TSP into one TSP stream

**Library**  DTV, Multiplex

**Class**  SDFDTV_SynThreeLayTSP

**Parameters**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Default</th>
<th>Type</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Length</td>
<td>number of bytes in one TSP</td>
<td>204</td>
<td>int</td>
<td>(204)</td>
</tr>
<tr>
<td>SegmentsA</td>
<td>number of segments in layer A</td>
<td>1</td>
<td>int</td>
<td>[1]†</td>
</tr>
<tr>
<td>LayerA_Modulation</td>
<td>modulation mode for layer A: A DQPSK, A QPSK, A 16QAM, A 64QAM</td>
<td>A DQPSK</td>
<td>enum</td>
<td></td>
</tr>
<tr>
<td>SegmentsB</td>
<td>number of segments in layer B</td>
<td>5</td>
<td>int</td>
<td>[1, 12]†</td>
</tr>
<tr>
<td>LayerB_Modulation</td>
<td>modulation mode for layer B: B DQPSK, B QPSK, B 16QAM, B 64QAM</td>
<td>B QPSK</td>
<td>enum</td>
<td></td>
</tr>
<tr>
<td>LayerB_Convolutional_Code</td>
<td>convolutional code rate for layer B: B 1/2, B 2/3, B 3/4, B 5/6, B 7/8</td>
<td>B 7/8</td>
<td>enum</td>
<td></td>
</tr>
<tr>
<td>SegmentsC</td>
<td>number of segments in layer C</td>
<td>7</td>
<td>int</td>
<td>[1, 12]†</td>
</tr>
<tr>
<td>LayerC_Modulation</td>
<td>modulation mode for layer C: C DQPSK, C QPSK, C 16QAM, C 64QAM</td>
<td>C 64QAM</td>
<td>enum</td>
<td></td>
</tr>
<tr>
<td>LayerC_Convolutional_Code</td>
<td>convolutional code rate for layer C: C 1/2, C 2/3, C 3/4, C 5/6, C 7/8</td>
<td>C 7/8</td>
<td>enum</td>
<td></td>
</tr>
</tbody>
</table>

† The sum of SegmentsA, SegmentsB and SegmentsC ≤ 13.
Pin Inputs

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>InA</td>
<td>layer A input</td>
<td>int</td>
</tr>
<tr>
<td>2</td>
<td>InB</td>
<td>layer B input</td>
<td>int</td>
</tr>
<tr>
<td>3</td>
<td>InC</td>
<td>layer B input</td>
<td>int</td>
</tr>
</tbody>
</table>

Pin Outputs

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>output</td>
<td>output TSP stream comprised of 3-layer TSP streams</td>
<td>int</td>
</tr>
</tbody>
</table>

Notes/Equations

1. This model is used in ISDB-T 3-layer systems to synthesize 3-layer TSP streams into one TSP stream, according to Table 5-3.

2. The model synthesizes the 3-layer TSP streams (layer A, layer B, and layer C) into one TSP stream.

   The number of TSPs per segment in each layer is determined according to modulation mode and convolutional code rate.

   The model determines the total number of TSPs in each layer by the number of TSPs per segments and the number of segments in each layer in one OFDM frame. To increase simulation speed, the total number of TSPs in each layer is divided by the greatest common divisor, which is the greatest common divisor of the total number of TSPs in each layer. An example follows for mode 1.

   SegmentsA=1
   LayerA_Modulation=A 64QAM
   LayerA_Convolutional_Code=A 2/3

   SegmentsB=5
   LayerB_Modulation=B DQPSK
   LayerB_Convolutional_Code=B 1/2

   SegmentsC=7
   LayerC_Modulation=C 16QAM
   LayerC_Convolutional_Code=C 3/4
Multiplex Components

Referring to Table 5-3, the number of TSPs is 48, 12, and 36 per segment, respectively. The total number of TSPs in each layer is 48×1, 12×5 and 36×7 in one OFDM frame; the greatest common divisor is 12. After dividing, the total number of TSPs is 4, 5 and 21, respectively. Then, the model combines 204×4×1 integers in InA pin, 204×1×5 integers in InB pin, and 204×3×7 integers in InC pin into the output TSP stream.

### Table 5-3. Transmitting TSPs per Segment for ISDB-T

<table>
<thead>
<tr>
<th>Carrier Modulation</th>
<th>Convolutional Code</th>
<th>Number of Transmitting TSPs† (Mode 1/2/3)</th>
</tr>
</thead>
<tbody>
<tr>
<td>DQPSK, QPSK</td>
<td>1/2</td>
<td>12/24/48</td>
</tr>
<tr>
<td></td>
<td>2/3</td>
<td>16/32/64</td>
</tr>
<tr>
<td></td>
<td>3/4</td>
<td>18/36/72</td>
</tr>
<tr>
<td></td>
<td>5/6</td>
<td>20/40/80</td>
</tr>
<tr>
<td></td>
<td>7/8</td>
<td>21/42/84</td>
</tr>
<tr>
<td>16-QAM</td>
<td>1/2</td>
<td>24/48/96</td>
</tr>
<tr>
<td></td>
<td>2/3</td>
<td>32/64/128</td>
</tr>
<tr>
<td></td>
<td>3/4</td>
<td>36/72/144</td>
</tr>
<tr>
<td></td>
<td>5/6</td>
<td>40/80/160</td>
</tr>
<tr>
<td></td>
<td>7/8</td>
<td>42/84/168</td>
</tr>
<tr>
<td>64-QAM</td>
<td>1/2</td>
<td>36/72/144</td>
</tr>
<tr>
<td></td>
<td>2/3</td>
<td>48/96/192</td>
</tr>
<tr>
<td></td>
<td>3/4</td>
<td>54/108/216</td>
</tr>
<tr>
<td></td>
<td>5/6</td>
<td>60/120/240</td>
</tr>
<tr>
<td></td>
<td>7/8</td>
<td>63/126/252</td>
</tr>
</tbody>
</table>

†number of transmitting TSPs per one OFDM frame

References

**DTV_SynTwoLayData**

**Description**  
Synthesizer for 2-layer data into one stream

**Library**  
DTV, Multiplex

**Class**  
SDFDTV_SynTwoLayData

**Parameters**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Default</th>
<th>Type</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Carriers</td>
<td>number of carriers in one segment</td>
<td>432</td>
<td>int</td>
<td>{96, 108,192,216,384,432}†</td>
</tr>
<tr>
<td>SegmentsA</td>
<td>number of segments in layer A</td>
<td>1</td>
<td>int</td>
<td>{1, 12}††</td>
</tr>
<tr>
<td>SegmentsB</td>
<td>number of segments in layer B</td>
<td>1</td>
<td>int</td>
<td>{1, 12}††</td>
</tr>
</tbody>
</table>

† According to ISDB-T, Carriers should be 96 and 108, 192 and 216, 384 and 432, corresponding to mode 1, mode 2, and mode 3, respectively.

†† <sup>sup</sup> The sum of SegmentsA and SegmentsB ≤ 13.

**Pin Inputs**

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>LayA</td>
<td>layer A input</td>
<td>complex</td>
</tr>
<tr>
<td>2</td>
<td>LayB</td>
<td>layer B input</td>
<td>complex</td>
</tr>
</tbody>
</table>

**Pin Outputs**

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>Synt</td>
<td>synthesis of 2-layer data streams</td>
<td>complex</td>
</tr>
</tbody>
</table>

**Notes/Equations**
Multiplex Components

1. The model is used in ISDB-T 2-layer systems to multiplex 2-layer data streams into one synthetic stream.

2. Each firing, LayA consumes $\text{Segments}_A \times \text{Carriers}$ tokens; LayB consumes $\text{Segments}_B \times \text{Carriers}$ tokens; Synt produces $(\text{Segments}_A + \text{Segments}_B) \times \text{Carriers}$ tokens.

References

**DTV_SynTwoLayTSP**

**Description**  Synthesizer for 2-layer TSP into one TSP stream  
**Library**  DTV, Multiplex  
**Class**  SDFDTV_SynTwoLayTSP

**Parameters**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Default</th>
<th>Type</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Length</td>
<td>number of bytes in one TSP</td>
<td>204</td>
<td>int</td>
<td>(204)</td>
</tr>
<tr>
<td>SegmentsA</td>
<td>number of segments in layer A</td>
<td>5</td>
<td>int</td>
<td>[1, 12]†</td>
</tr>
<tr>
<td>LayerA_Modulation</td>
<td>modulation mode for layer A: A DQPSK, A QPSK, A 16QAM, A 64QAM</td>
<td>A DQPSK</td>
<td>enum</td>
<td></td>
</tr>
<tr>
<td>SegmentsB</td>
<td>number of segments in layer B</td>
<td>8</td>
<td>int</td>
<td>[1, 12]†</td>
</tr>
<tr>
<td>LayerB_Modulation</td>
<td>modulation mode for Layer B: B DQPSK, B QPSK, B 16QAM, B 64QAM</td>
<td>B DQPSK</td>
<td>enum</td>
<td></td>
</tr>
<tr>
<td>LayerB_Convolutional_Code</td>
<td>convolutional code rate for layer B: B 1/2, B 2/3, B 3/4, B 5/6, B 7/8</td>
<td>B 7/8</td>
<td>enum</td>
<td></td>
</tr>
</tbody>
</table>

† The sum of SegmentsA and SegmentsB ≤ 13.

**Pin Inputs**

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>InA</td>
<td>layer A input</td>
<td>int</td>
</tr>
<tr>
<td>2</td>
<td>InB</td>
<td>layer B input</td>
<td>int</td>
</tr>
</tbody>
</table>

DTV_SynTwoLayTSP  5-29
Multiplex Components

Pin Outputs

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>output</td>
<td>output TSP stream comprised of 2-layer TSP streams</td>
<td>int</td>
</tr>
</tbody>
</table>

Notes/Equations

1. This model is used in ISDB-T 2-layer systems to synthesize a 2-layer TSP stream into one TSP stream according to Table 5-4.

2. The model synthesizes layer A and layer B into one TSP stream.

   The number of TSPs per segment in each layer is determined according to modulation mode and convolutional code rate.

   The model determines the total number of TSPs in each layer by the number of TSPs per segment and the number of segments in each layer in one OFDM frame. To increase simulation speed, the total number of TSPs in each layer is divided by the greatest common divisor, which is the greatest common divisor of the total numbers of TSPs in each layer. An example follows for mode 1.

   SegmentsA = 5
   LayerA_Modulation = A DQPSK
   LayerA_Convolutional_Code = A 1/2

   SegmentsB = 7
   LayerB_Modulation = B 16QAM
   LayerB_Convolutional_Code = B 3/4

   According to Table 5-4, the number of TSPs is 12, and 36 per segment, respectively. The total number of TSPs in each layer is 12x5 and 36x7 in one OFDM frame; the greatest common divisor is 12. After dividing, the total number of TSPs is 5 and 21, respectively. Then, the model combines 204x5x1 integers in InA pin and 204x3x7 integers in InB pin into the output TSP stream.

Table 5-4. Transmitting TSPs per Segment for ISDB-T

<table>
<thead>
<tr>
<th>Carrier Modulation</th>
<th>Convolutional Code</th>
<th>Number of Transmitting TSPs† (Mode 1/2/3)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

5-30  DTV_SynTwoLayTSP
Table 5-4. Transmitting TSPs per Segment for ISDB-T

<table>
<thead>
<tr>
<th>Modulation</th>
<th>1/2</th>
</tr>
</thead>
<tbody>
<tr>
<td>DQPSK, QPSK</td>
<td>12/24/48</td>
</tr>
<tr>
<td>2/3</td>
<td>16/32/64</td>
</tr>
<tr>
<td>3/4</td>
<td>18/36/72</td>
</tr>
<tr>
<td>5/6</td>
<td>20/40/80</td>
</tr>
<tr>
<td>7/8</td>
<td>21/42/84</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Modulation</th>
<th>1/2</th>
</tr>
</thead>
<tbody>
<tr>
<td>16-QAM</td>
<td>24/48/96</td>
</tr>
<tr>
<td>2/3</td>
<td>32/64/128</td>
</tr>
<tr>
<td>3/4</td>
<td>36/72/144</td>
</tr>
<tr>
<td>5/6</td>
<td>40/80/160</td>
</tr>
<tr>
<td>7/8</td>
<td>42/84/168</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Modulation</th>
<th>1/2</th>
</tr>
</thead>
<tbody>
<tr>
<td>64-QAM</td>
<td>36/72/144</td>
</tr>
<tr>
<td>2/3</td>
<td>48/96/192</td>
</tr>
<tr>
<td>3/4</td>
<td>54/108/216</td>
</tr>
<tr>
<td>5/6</td>
<td>60/120/240</td>
</tr>
<tr>
<td>7/8</td>
<td>63/126/252</td>
</tr>
</tbody>
</table>

† number of transmitting TSPs per one OFDM frame

References

Multiplex Components
Chapter 6: OFDM Components
OFDM Components

**DTV_AddFixPhase**

**Description**  
Fixed phase addition to the OFDM symbol

**Library**  
DTV, OFDM

**Class**  
SDFDTV_AddFixPhase

**Parameters**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Default</th>
<th>Type</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Length</td>
<td>length of input symbol</td>
<td>8192</td>
<td>int</td>
<td>$(0, \infty)$†</td>
</tr>
<tr>
<td>Guard</td>
<td>length of guard interval</td>
<td>512</td>
<td>int</td>
<td>$(0, \text{Length})$††</td>
</tr>
<tr>
<td>Offset</td>
<td>fixed phase offset</td>
<td>0.25</td>
<td>real</td>
<td>$(-0.5, 0.5)$</td>
</tr>
</tbody>
</table>

† Length = $2^N$ in OFDM systems; where $N$ is a positive integer.  
In DVB-T systems, $N=11$ and 13; in ISDB-T systems, $N=11, 12$ and 13.

†† Guard = $1/32, 1/16, 1/8$, and $1/4$ Length in ISDB-T and DVB-T systems.

**Pin Inputs**

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>input</td>
<td>OFDM symbol after insert guard interval process</td>
<td>complex</td>
</tr>
</tbody>
</table>

**Pin Outputs**

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>output</td>
<td>symbol added by a fixed phase offset</td>
<td>complex</td>
</tr>
</tbody>
</table>

**Notes/Equations**

1. This model is used to add a fixed phase offset into the OFDM symbol; it can be used as a tool for frequency synchronization models.

2. Implementation
Assume \( x(0), x(1), \ldots, x(N-1) \) are the transmitted symbols that are combined into an OFDM symbol. Add a fixed phase offset \( \Phi \) into OFDM symbol, we can get \( y(0), y(1), \ldots, y(N-1) \), as follows:

\[
y(i) = x(i)e^{j2\pi\Phi N}
\]

where \( \Phi \) is the carrier phase offset, its value is in \((-0.5, 0.5)\) set. We add the fixed phase offset in the \( \pm 0.5 \) subcarriers in OFDM symbol. \( N \) is the sum of the length of IFFT and the length of the interval guard.

References


OFDM Components

**DTV_InsertGuard**

**Description**  
Guard interval inserter

**Library**  
DTV, OFDM

**Class**  
SDFDTV_InsertGuard

**Parameters**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Default</th>
<th>Type</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Length</td>
<td>length of input symbol</td>
<td>8192</td>
<td>int</td>
<td>((0, \infty))†</td>
</tr>
<tr>
<td>Guard</td>
<td>length of guard interval</td>
<td>512</td>
<td>int</td>
<td>((0, \text{Length}))††</td>
</tr>
</tbody>
</table>

† Length = \(2^N\) in OFDM systems; where N is a positive integer.  
In DVB-T systems, N=11 and 13; in ISDB-T systems, N=11, 12 and 13.

††Guard = 1/32, 1/16, 1/8, and 1/4 Length in ISDB-T and DVB-T systems.

**Pin Inputs**

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>input</td>
<td>signal from the IFFT</td>
<td>complex</td>
</tr>
</tbody>
</table>

**Pin Outputs**

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>output</td>
<td>signal output after guard interval insertion</td>
<td>complex</td>
</tr>
</tbody>
</table>

**Notes/Equations**

1. This model is used to insert the guard interval after the IFFT procedure.

2. Model implementation
Assume \( x(0), x(1), \ldots, x(N - 1) \) are the \( N \) symbols from IFFT. The guard interval (interval length is \( N_g \)) is inserted as follows:

\[
x(-i) = x(N - i) \quad i = 1, 2, \ldots, N_g
\]

After insertion of the guard interval, the length of the output signal is \( N + N_g \), the output signal is

\[
x(-N_g), \ldots, x(-1), x(0), x(1), \ldots, x(N - 1)
\]

Guard interval insertion is illustrated in Figure 6-1.

![Figure 6-1. Guard Interval Insertion](image)
OFDM Components

DTV_LoadFFTBuff

Description  Received data loader from channel to FFT buffer
Library   DTV, OFDM
Class   SDFDTV_LoadFFTBuff

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Default</th>
<th>Type</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>InLength</td>
<td>length of input sequence</td>
<td>2048</td>
<td>int</td>
<td>[0, ∞)†</td>
</tr>
<tr>
<td>Order</td>
<td>FFT points=2^Order</td>
<td>11</td>
<td>int</td>
<td>[1, ∞)††</td>
</tr>
<tr>
<td>MinDelay</td>
<td>min delay from 0 to InLength-1</td>
<td>0</td>
<td>int</td>
<td>(0, InLength]</td>
</tr>
<tr>
<td>MaxDelay</td>
<td>max delay from 0 to InLength-1</td>
<td>2047</td>
<td>int</td>
<td>[MinDelay, InLength]</td>
</tr>
<tr>
<td>Offset</td>
<td>offset from ML peak to symbol start point</td>
<td>256</td>
<td>int</td>
<td>[-MinDelay, 2^Order-1]†††</td>
</tr>
</tbody>
</table>

† InLength = 2^N in OFDM systems; where N is a positive integer. N=11 and 13 in DVB-T systems; N=11, 12, and 13 in ISDB-T systems.
†† Order=11 and 13 in DVB-T systems, Order=11, 12 and 13 in ISDB-T systems.
††† Offset = 1/32, 1/16, 1/8, and 1/4 InLength in ISDB-T and DVB-T systems.

Pin Inputs

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>input</td>
<td>input signal from channel</td>
<td>complex</td>
</tr>
<tr>
<td>2</td>
<td>corr</td>
<td>ML estimation of theta for OFDM symbol synchronization</td>
<td>real</td>
</tr>
<tr>
<td>3</td>
<td>angle</td>
<td>phase offset corresponding to the theta</td>
<td>real</td>
</tr>
</tbody>
</table>
Pin Outputs

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>output</td>
<td>output signal which be used by FFT</td>
<td>complex</td>
</tr>
<tr>
<td>5</td>
<td>phase</td>
<td>phase offset of the current OFDM symbol</td>
<td>real</td>
</tr>
</tbody>
</table>

Notes/Equations

1. This model is used to perform OFDM symbol synchronization, determine phase offset due to frequency offset, then output the OFDM symbol (output data is $2^{\text{Order}}$) to the FFT buffer.

2. Implementation

Assume $2(N+L)$ consecutive samples of $x(k)$ is observed, Figure 6-2, and that these samples contain one complete $(N+L)$ sample OFDM symbol.

![Figure 6-2. Structure of OFDM Signal with Cyclically Extended Symbols, $s(k)$](image)

The position of this symbol within the observed block of samples, however, is unknown as the channel delay $\theta$ is unknown to the receiver. Define the index sets $I \equiv [\theta, \theta + L - 1]$ and $I' \equiv [\theta + N, \theta + N + L - 1]$ (see Figure 6-2). The sets $I'$ thus contain the indices of data samples that are copied into the cyclic extension, and the set $I$ contains the indices of this extension. Collect the observed samples in the $2(N+L)$-vector $X \equiv [x(1) \ldots x(2(N + L))]^T$. Note that the samples in the cyclic extension and their replicas, $k \in I \cup I'$ are pair wise correlated, i.e., $\forall k \in I$: 

DTV_LoadFFTBuff 6-7
OFDM Components

\[
E \{ x(k)x^*(k+m) \} = \begin{cases} 
\sigma_s^2 + \sigma_n^2 & m = 0 \\
\sigma_s^2 e^{j2\pi \epsilon} & m = N \\
0 & \text{otherwise}
\end{cases}
\]

where

\[
\sigma_s^2 = E \{ |s(k)|^2 \} \quad \text{and} \quad \sigma_n^2 = E \{ |n(k)|^2 \},
\]

while the remaining samples, \( k \notin 1 \cup I' \), are mutually not correlated. We now explicitly exploit this correlation property and give the simultaneous maximum likelihood (ML) estimates of \( \theta \) and \( \epsilon \).

The log-likelihood function for \( \theta \) and \( \epsilon \) is the logarithm of the probability density function of the \( 2(N+L) \) observed samples in \( X \) given the arrival time \( \theta \) and the carrier frequency offset \( \epsilon \). The ML estimation of \( \theta \) and \( \epsilon \) is the argument maximizing this function. Under the assumption that \( X \) is a jointly Gaussian vector, the log-likelihood function can be shown to be

\[
\Lambda(\theta, \epsilon) = 2|\gamma(\theta)| \cos \left( \{ 2\pi \epsilon + \angle \gamma(\theta) \} - \rho \Phi(\theta) \right)
\]

where \( \angle \) denotes the argument of a complex number,

\[
\gamma(m) = \sum_{k=m}^{m+L-1} x(k)x^*(k+N)
\]

\[
\epsilon(m) = \sum_{k=m}^{m+L-1} |x(k)|^2 + |x(k+N)|^2
\]

are a correction term and an energy term, and

\[
\rho = \frac{\sigma_s^2}{\sigma_s^2 + \sigma_n^2}
\]

is the magnitude of the correction coefficient between \( x(k) \) and \( x(k+N) \).
The simultaneous ML estimation of $\theta$ and $\epsilon$ becomes

$$\hat{\theta}_{\text{ML}} = \arg \max_{\theta} \{2|\gamma(\theta)| - \rho \Phi(\theta)\}$$

$$\hat{\epsilon}_{\text{ML}} = \frac{1}{2\pi} \angle(\hat{\theta}_{\text{ML}})$$

The processing is done continuously and two signals are generated. $\hat{\theta}_{\text{ML}}$ is the start point for current OFDM symbol; $\hat{\epsilon}_{\text{ML}}$ is the phase offset in the current OFDM symbol.

In this model, two signals $2|\gamma(\theta)| - \rho \Phi(\theta)$ and its corresponding phase offset signal

$$\epsilon = \frac{1}{2\pi} \angle(\theta)$$

were generated in the DTV_MLEstimator model, these two signals are input pins (corr and angle) of this model. This model determines the maximum value $\hat{\theta}_{\text{ML}}$ of the corr pin in the range [MinDelay, MaxDelay], then determines the current carrier frequency offset $\hat{\epsilon}_{\text{ML}}$ corresponding to the $\hat{\theta}_{\text{ML}}$; then, according to time $\hat{\theta}_{\text{ML}}$, determines the N points signal to the output pin.

References


OFDM Components

**DTV_MLEstimator**

**Description**  
ML Estimation and Synchronization of OFDM Symbol

**Library**  
DTV, OFDM

**Class**  
SDFDTV_MLEstimator

**Parameters**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Default</th>
<th>Type</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Length</td>
<td>length of OFDM symbol</td>
<td>8192</td>
<td>int</td>
<td>$(0, \infty) \dagger$</td>
</tr>
<tr>
<td>Guard</td>
<td>length of guard interval</td>
<td>256</td>
<td>int</td>
<td>$(0, \text{Length}) \dagger\dagger$</td>
</tr>
<tr>
<td>Ru</td>
<td>scale of the square term in ML algorithm</td>
<td>0.95</td>
<td>real</td>
<td>$(0.0, 1.0)$</td>
</tr>
</tbody>
</table>

$\dagger$ Length = $2^N$ in OFDM systems; where $N$ is a positive integer.  
In DVB-T systems, $N=11$ and 13; in ISDB-T systems, $N=11, 12$ and 13.

$\dagger\dagger$ Guard = 1/32, 1/16, 1/8, and 1/4 Length in ISDB-T and DVB-T systems.

**Pin Inputs**

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>input</td>
<td>input signal from channel</td>
<td>complex</td>
</tr>
</tbody>
</table>

**Pin Outputs**

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>corr</td>
<td>ML estimation of theta for OFDM symbol synchronization</td>
<td>real</td>
</tr>
<tr>
<td>3</td>
<td>angle</td>
<td>phase offset corresponding to the theta</td>
<td>real</td>
</tr>
</tbody>
</table>
Notes/Equations

1. This model is used to calculate the parameters for simultaneous ML estimation of $\theta$ and $\varepsilon$, which is used in timing and frequency synchronization in OFDM systems. The outputs of this model will be used by DTV_LoadFFTBuff.

2. Implementation

The $\theta$ and $\varepsilon$ are defined in DTV_LoadFFTBuff. The log-likelihood function can be shown to be

$$\Lambda(\theta, \varepsilon) = 2|\gamma(\theta)|\cos\{(2\pi\varepsilon + \angle\gamma(\theta)) - \rho\Phi(\theta)\}$$

where $\angle$ denotes the argument of a complex number;

$$\gamma(m) = \sum_{k=m}^{m+L-1} x(k)x^*(k+N)$$

$$\Phi(m) = \sum_{k=m}^{m+L-1} |x(k)|^2 + |x(k+N)|^2$$

are correction and energy terms, and

$$\rho = \frac{\sigma^2_s}{\sigma^2_s + \sigma^2_n}$$

is the magnitude of the correction coefficient between $x(k)$ and $x(k+N)$. The simultaneous ML-estimation of $\theta$ and $\varepsilon$ becomes

$$\hat{\theta}_{ML} = \arg\max_{\theta} \{2|\gamma(\theta)| - \rho\Phi(\theta)\}$$

$$\hat{\varepsilon}_{ML} = \frac{1}{2\pi}\angle\gamma(\hat{\theta}_{ML})$$

In order to get the above two values for the OFDM symbol, two signals are needed:

$$2|\gamma(\theta)| - \rho\Phi(\theta)$$

and its corresponding phase offset signal.
OFDM Components

\[ \varepsilon = -\frac{1}{2\pi} \angle \gamma(\theta) \]

This model generates both signals. The output of the corr pin is the value of

\[ 2|\gamma(\theta)| - \rho \varepsilon(\theta); \]

and the output of the angle is the value of

\[ \varepsilon = -\frac{1}{2\pi} \angle \gamma(\theta). \]

References


**DTV_OFDMEqualizer**

**Description**  
OFDM equalizer by the channel estimation

**Library**  
DTV, OFDM

**Class**  
SDFDTV_OFDMEqualizer

### Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Default</th>
<th>Type</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Carriers</td>
<td>number of active carriers in one OFDM symbol</td>
<td>1705</td>
<td>int</td>
<td>(0, ∞)†</td>
</tr>
</tbody>
</table>

† In ISDB-T systems: Carriers = \( n \times 108 \) in mode 1, \( n \times 216 \) in mode 2, or \( n \times 384 \) in mode 3, where \( n \) is the number of segments.

In DVB-T systems: Carriers = 1705 in 2k mode, or 6817 in 8k mode.

### Pin Inputs

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>input</td>
<td>data in the active carriers in OFDM symbol</td>
<td>complex</td>
</tr>
<tr>
<td>2</td>
<td>Coef</td>
<td>frequency channel impulse response (CIR) estimation</td>
<td>complex</td>
</tr>
</tbody>
</table>

### Pin Outputs

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>output</td>
<td>output data after channel equalization</td>
<td>complex</td>
</tr>
</tbody>
</table>

### Notes/Equations

1. The model is used to perform channel equalization by the channel estimation in each of the active carriers.

2. Implementation
OFDM Components

The DTV_ChEstimator and DTV_DVBChEstimator models provide channel estimation and received signal in each active carrier. The OFDM channel equalization algorithm is:

\[ a(i) = \frac{x(i)}{h(i)} \]

where \( h(i) \) is the channel estimation, \( x(i) \) is the received signal in active carriers, \( a(i) \) is the equalized output signal.

References


**DTV_RemovePhase**

*Description*  
Compensator for phase offset due to carrier frequency offset  

*Library*  
DTV, OFDM  

*Class*  
SDFDTV_RemovePhase  

**Parameters**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Default</th>
<th>Type</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Length</td>
<td>length of input symbol</td>
<td>8192</td>
<td>int</td>
<td>(0, ∞)†</td>
</tr>
</tbody>
</table>

† Length = $2^N$ in OFDM systems; where N is a positive integer.  
In DVB-T systems, N=11 and 13; in ISDB-T systems, N=11, 12 and 13.

**Pin Inputs**

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>input</td>
<td>signal from the OFDM symbol after OFDM symbol synchronization</td>
<td>complex</td>
</tr>
<tr>
<td>2</td>
<td>phase</td>
<td>phase offset due to the carrier frequency offset</td>
<td>real</td>
</tr>
</tbody>
</table>

**Pin Outputs**

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>output</td>
<td>OFDM signal after removed the phase offset</td>
<td>complex</td>
</tr>
</tbody>
</table>

**Notes/Equations**

1. This model is used to remove the random phase offset due to the carrier frequency offset in the OFDM symbol before OFDM channel estimation.
2. Implementation
OFDM Components

Assume \( x(0), x(1), \ldots, x(N-1) \) are the received signal after OFDM synchronization. The phase \( \Phi \) is the detected phase offset due to the carrier frequency offset in DTV_LoadFFTBuffer model. The phase removed OFDM signal \( y(0), y(1), \ldots, y(N-1) \) is:

\[
y(i) = x(i)e^{-j2\pi \Phi / N}
\]

References


Chapter 7: Test Components
Test Components

DTV_BER

Description  Bit error rate for ISDB-T and DVB-T
Library   DTV, Test
Class   SDFDTV_BER

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Default</th>
<th>Type</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Length</td>
<td>length of input byte block</td>
<td>188</td>
<td>int</td>
<td>$[1, \infty)$</td>
</tr>
<tr>
<td>Delay</td>
<td>delay of byte blocks</td>
<td>512</td>
<td>int</td>
<td>$[0, \infty)$</td>
</tr>
</tbody>
</table>

Pin Inputs

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>test</td>
<td>test input bit</td>
<td>int</td>
</tr>
<tr>
<td>2</td>
<td>ref</td>
<td>reference input bit</td>
<td>int</td>
</tr>
</tbody>
</table>

Pin Outputs

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>ber</td>
<td>bit error rate</td>
<td>real</td>
</tr>
</tbody>
</table>

Notes/Equations

1. This model is used to measure the input signal bit error rate.

   Each firing, 1 token at pin ber is produced when Length×8 tokens are consumed (note that Length is bytes, while test and ref are bits).

2. If input test bit is $t_i$, input reference bit is $r_i$, the model calculates the error rate after Delay according to the following example.

7-2  DTV_BER
count=0;
err =0;
rate =0;
if (count>Delay) {
    for (i=0;i<length*8;i++) {
        t_i = test[i];
        r_i = ref[i];
        t_i = t_i & 1;
        r_i = r_i & 1;
        if (t_i != r_i )
            err = err+1;
    }
    count++;
    if (count>Delay)
        rate = err/((count-Delay)*length*8);
    BER%(0) <<rate;
}
Test Components

**DTV_PowerMeasure**

**Description**  Average power measurement

**Library**  DTV, Test

### Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Default</th>
<th>Sym</th>
<th>Type</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>BlockSize</td>
<td>number of particles in a block</td>
<td>16</td>
<td>m</td>
<td>int</td>
<td>[1, ∞)</td>
</tr>
</tbody>
</table>

### Pin Inputs

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>input</td>
<td>input signal</td>
<td>real</td>
</tr>
</tbody>
</table>

### Pin Outputs

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>Ave_P</td>
<td>average power of all input signals</td>
<td>real</td>
</tr>
<tr>
<td>3</td>
<td>B_P</td>
<td>average power of input signals in a block</td>
<td>real</td>
</tr>
</tbody>
</table>

### Notes/Equations

1. This subnetwork model is used to measure the average power of the input signals.

   Each firing, 1 token at Blk_P and Avg_P are produced when \( m \) input tokens are consumed.

2. If input signal \( s(t) = x(t) + jy(t) \),

   \[
   \text{average signal power } P = \int_{a}^{b} (x^2(t) + y^2(t)) \, dt
   \]
for discrete signals, \( P = \frac{\sum_{n=a}^{b} (x^2(n) + y^2(n))}{b-a} \)

For Blk_P out, \( b - a = m \); for Avg_P, \( a = 0, \) \( b = \) current input points.

The schematic for this subnetwork is shown in Figure 7-1.

Figure 7-1. DTV_PowerMeasure Schematic
Test Components
Chapter 8: DVB-T Design Examples

Introduction

DTV example designs can be accessed from the /examples/dtv directory. Schematics and simulation results for the following design examples are described in this chapter.

DTV_DVBOFDM_prj
- DsnDTV_DVBOFDM_16QAM.dsn
- DsnDTV_DVBOFDM_64QAM.dsn
- DsnDTV_DVBOFDM_PALInterference.dsn

DTV_DVBSystem_prj
- DsnDTV_DVBSystem_16QAM.dsn
- DsnDTV_DVBSystem_Hier64QAM.dsn
- DsnDTV_DVBOFDM_16QAM_BER.dsn
OFDM 16-QAM Modulation and Demodulation in DVB-T Systems

DTV_DVBOFDM_prj Design Name

DsnDTV_DVBOFDM_16QAM.dsn

Features

• 16-QAM modulation and demodulation

• TDMA multipath fading channel with Doppler shift. The type of multipath can be selected; Doppler shift can be determined by the mobile speed setting.

• Comparison of DVB and TDMA channel system performance

Description

This example design provides modulation, transmission, and demodulation via a TDMA channel with Doppler shift and DVB channel. Modulation mapping is 16-QAM; the guard interval ratio is 1/16.

After 16-QAM data mapping, OFDM symbols are formed by adding TPS data and pilots. 2048-point inverse FFT (IFFT) is performed. After inserting the guard interval, the complex signal is transmitted; the transmitted signal length is 2048+guard interval.

In the receiver, the FFT starting point is determined by the autocorrection function of the received signal. After FFT, the transmitted signal is recovered by the frequency equalizer. According to the DVB-T, the mapped data is received. Data is recovered by the 16-QAM demodulator.

Performance can be viewed in data display files such as DsnDTV_DVBOFDM_16QAM and DsnDTV_DVBOFDM_16QAM_EVM.

Schematics

Figure 8-1 shows the schematic for this design. Three subnetworks are used in this design; they are shown in Figure 8-2 through Figure 8-4.
DVB-T Design Examples

Figure 8-1. DsnDTV_DVBOFDM_16QAM.dsn

Figure 8-2. sub_DVBOFDM_Mod.dsn

Figure 8-3. sub_DVBOFDM_Demod.dsn

8-4 DTV OFDM
Specifications

<table>
<thead>
<tr>
<th>Symbol (Model)</th>
<th>Specification (Parameter)</th>
<th>Simulation Type</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>RxSpectrum</td>
<td>Stop</td>
<td>ADS Ptolemy</td>
<td>$224 \times (2048 + 128)/2048 \times 2\ \mu\text{sec}$</td>
</tr>
<tr>
<td>RxSignal</td>
<td>Stop</td>
<td>ADS Ptolemy</td>
<td>$224 \times (2048 + 128)/2048 \ \mu\text{sec}$</td>
</tr>
<tr>
<td>PropNADCtdma</td>
<td>Type</td>
<td>ADS Ptolemy</td>
<td>TwoPath</td>
</tr>
<tr>
<td>PropNADCtdma</td>
<td>Pathloss</td>
<td>ADS Ptolemy</td>
<td>No</td>
</tr>
<tr>
<td>PropNADCtdma</td>
<td>Env</td>
<td>ADS Ptolemy</td>
<td>TypicalSuburban</td>
</tr>
<tr>
<td>PropNADCtdma</td>
<td>Delay</td>
<td>ADS Ptolemy</td>
<td>0.5 $\mu\text{sec}$</td>
</tr>
<tr>
<td>PropNADCtdma</td>
<td>Test</td>
<td>ADS Ptolemy</td>
<td>Tap1</td>
</tr>
<tr>
<td>AntMobile</td>
<td>Vx</td>
<td>ADS Ptolemy</td>
<td>100.0 km/hr</td>
</tr>
<tr>
<td>AntMobile</td>
<td>Vy</td>
<td>ADS Ptolemy</td>
<td>0.0 km/hr</td>
</tr>
</tbody>
</table>

Notes

1. In this example, the guard interval ratio is 1/16, so the guard interval is 128. (According to DVB-T, in the 2k mode the guard interval is 64, 128, 256, or 512, which corresponds to 1/32, 1/16, 1/8, or 1/4 guard interval ratio, respectively.)

2. All parameters are related to the guard interval (except TDMA channel parameters). If the guard interval is changed, parameters will be changed to the corresponding values.
3. The TDMA channel condition varies according to channel parameter modification. Parameters include Type, Pathloss, Env, Delay, and Test in the PropNADCtdma model, and Vx and Vy in the AntMobile model.

Simulation Results

Figure 8-5. ML magnitude, which shows the magnitude of ML function of received signal used to find the FFT start. The maximum value appears in every (2048+128) points, the point of the maximum magnitude value is the FFT start.

Figure 8-6. Received constellation after OFDM demodulation via DVB channel. Results show the hard decision before the 16-QAM demodulation is correct.
Figure 8-7. Received constellation after OFDM demodulation via TDMA channel. Results show OFDM system performance.

Figure 8-8. Spectrum of signal received from wireless channel. Center frequency is 474 MHz

Figure 8-9. Adjacent-Channel Power Ratio
The equation is:
\[
\text{ACPR} = \text{acpr\_vr(DsnDTV\_DVBOFDM\_16QAM..RxSignal,50.0,}
\{-3.9MHz,3.9MHz\},
\{-12MHz,-4.25MHz\},
\{4.25MHz,12MHz\})
\]
Figure 8-10. Relative magnitude error, which is defined as:
\[ EVM = \frac{\text{mag}(\text{DsnDTV_DVBOFDM_16QAM..Rx16QAM1-DsnDTV_DVBOFDM_16QAM..Tx16QAM})}{\text{mean}(\text{mag}(\text{DsnDTV_DVBOFDM_16QAM..Tx16QAM}))} \]

Figure 8-11. EVM histogram, which shows most EVM < 0.1.
EVMhist is defined as:
\[ \text{EVMhist} = \text{histogram}(\text{EVM,1001,0,0,1,0}) \]

Benchmark
- Hardware Platform: Pentium II 200 MHz, 96 MB memory
- Software Platform: Windows NT 4.0 Workstation, ADS 1.3
- Data Points: 10 OFDM symbols
- Simulation Time: 34 seconds

References
OFDM 64-QAM Modulation and Demodulation in DVB-T Systems

DTV_DVBOFDM_prj Design Name
DsnDTV_DVBOFDM_64QAM.dsn

Features

- 64-QAM modulation and demodulation
- TDMA multipath fading channel with Doppler shift. The type of multipath can be selected; Doppler shift can be determined by the mobile speed setting.
- DVB and TDMA channel performance comparison

Description

This example design provides modulation, transmission and demodulation via a TDMA channel with Doppler shift and DVB channel. Modulation mapping is 64-QAM; the guard interval ratio is 1/16.

After 64-QAM data mapping, OFDM symbols are formed by adding TPS data and pilots. 2048-point inverse FFT (IFFT) is performed. After inserting the guard interval, the complex signal is transmitted. Transmitted signal length is 2048+guard interval.

In the receiver, the FFT starting point is determined by the autocorrection function of the received signal. After FFT, the transmitted signal is recovered by the frequency equalizer. According to the DVB-T, the mapped data is received. Data is recovered by the 64-QAM demodulator.

Performance can be viewed in the data display files such as DsnDTV_DVBOFDM_64QAM and DsnDTV_DVBOFDM_64QAM_EVM.

Schematics

Figure 8-12 shows the schematic for this design. Three subnetworks are used in this design; they are shown in Figure 8-13 through Figure 8-15.
DVB-T Design Examples

Figure 8-12. DsnDTV_DVBOFDM_64QAM.dsn

Figure 8-13. sub_DVBOFDM_Mod.dsn

Figure 8-14. sub_DVBOFDM_Demod.dsn

8-10  DTV OFDM
Specifications

<table>
<thead>
<tr>
<th>Symbol (Model)</th>
<th>Specification (Parameter)</th>
<th>Simulation Type</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>RxSpectrum</td>
<td>Stop</td>
<td>ADS Ptolemy</td>
<td>224*(2048+128)/2048*2 µsec</td>
</tr>
<tr>
<td>RxSignal</td>
<td>Stop</td>
<td>ADS Ptolemy</td>
<td>224*(2048+128)/2048 µsec</td>
</tr>
<tr>
<td>PropNADCtdma</td>
<td>Type</td>
<td>ADS Ptolemy</td>
<td>TwoPath</td>
</tr>
<tr>
<td>PropNADCtdma</td>
<td>Pathloss</td>
<td>ADS Ptolemy</td>
<td>No</td>
</tr>
<tr>
<td>PropNADCtdma</td>
<td>Env</td>
<td>ADS Ptolemy</td>
<td>Typical/Suburban</td>
</tr>
<tr>
<td>PropNADCtdma</td>
<td>Delay</td>
<td>ADS Ptolemy</td>
<td>2.5 µsec</td>
</tr>
<tr>
<td>AntMobile</td>
<td>Vx</td>
<td>ADS Ptolemy</td>
<td>30.0 km/hr</td>
</tr>
<tr>
<td>AntMobile</td>
<td>Vy</td>
<td>ADS Ptolemy</td>
<td>0.0 km/hr</td>
</tr>
</tbody>
</table>

Notes

1. In this example, the guard interval ratio is 1/16, so the guard interval is 128. (According to DVB-T, in the 2k mode the guard interval is 64, 128, 256, or 512, which corresponds to 1/32, 1/16, 1/8, or 1/4 guard interval ratio, respectively.)

2. All parameters are related to the guard interval (except TDMA channel parameters). If the guard interval is changed, parameters will be changed to the corresponding values.
3. The TDMA channel condition varies according to channel parameter modification. Parameters include Type, Pathloss, Env, Delay, and Test in the PropNADCtdma model, and Vx and Vy in the AntMobile model.

Simulation Results

Figure 8-16. ML (maximum likelihood) magnitude which shows the magnitude of ML function of the received signal used to find the FFT start point. The maximum value appears in every (2048+128) points, the point of the maximum magnitude value is the start point of the FFT.

Figure 8-17. Received constellation after OFDM demodulation via TDMA channel. Results show the hard decision before the 64-QAM demodulation is correct.
Figure 8-18. Received constellation after OFDM demodulation via DVB channel. Results show OFDM system performance.

Figure 8-19. Spectrum of signal received from wireless channel. Center frequency is 474 MHz

Figure 8-20. Adjacent-Channel Power Ratio.
The equation is:
\[
ACPR = \text{acpr}_v(DsnDTV_DVBOFDM_64QAM..RxSignal, 50.0, \{-3.9MHz,3.9MHz\}, \{-12MHz,-4.25MHz\}, \{4.25MHz,12MHz\})
\]
Figure 8-21. Relative magnitude error, which is defined as:

\[
EVM = \frac{\text{mag(DsnDTV \_ DVBOFDM \_ 64QAM \_ Rx64QAM \_ 1 - DsnDTV \_ DVBOFDM \_ 64QA \_ M \_ Tx64QAM)}}{\text{mean(mag(DsnDTV \_ DVBOFDM \_ 64QAM \_ M \_ Tx64QAM))}}
\]

Figure 8-22. EVM histogram, which shows most EVM < 0.1.

The EVM hist is defined as:

\[
EVM\text{hist} = \text{histogram(EVM,1001,0,0,1.0)}
\]
Benchmark

- Hardware Platform: Pentium II 200 MHz, 96 MB memory
- Software Platform: Windows NT 4.0 Workstation, ADS 1.3
- Data Points: 10 OFDM symbols
- Simulation Time: 34 seconds

References

OFDM, 64-QAM DTV and PAL Signal Interference Test in DTV Systems

DTV_DVBOFDM_prj Design Name
DsnDTV_DVBOFDM_PALInterference.dsn

Features
- 64-QAM modulation and demodulation
- Adjacent channel interference between analog PAL TV signal and digital DVB-T signal
- Compares analog PAL TV signal and received PAL TV signal that has adjacent channel DVB-T signal interference
- Constellation of received DVB-T signal with adjacent channel PAL signal interference

Description
This design is an OFDM example that tests the adjacent channel interference between the analog PAL TV signal and the digital DVB-T signal. The DVB-T signal has weak interference but the analog TV signal experiences strong interference. This effect is saved Data Display (DsnDTV_DVBOFDM_PALInterference1.dds). The center frequency of DVB-T is 482 MHz. The PAL spectrum experiences adjacent channel interference at higher frequency.

Schematics
Figure 8-23 shows the schematic for this design. Two subnetworks are used in this design; they are shown in Figure 8-24 and Figure 8-25.
Specifications

<table>
<thead>
<tr>
<th>Specification (Parameter)</th>
<th>Simulation Type</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>IFFTSize</td>
<td>ADS Ptolemy</td>
<td>2048*4</td>
</tr>
<tr>
<td>FFTSize</td>
<td>ADS Ptolemy</td>
<td>2048*4</td>
</tr>
<tr>
<td>GuardInterval</td>
<td>ADS Ptolemy</td>
<td>128*4</td>
</tr>
<tr>
<td>IFFTOrder</td>
<td>ADS Ptolemy</td>
<td>13</td>
</tr>
<tr>
<td>FFTOrder</td>
<td>ADS Ptolemy</td>
<td>13</td>
</tr>
<tr>
<td>MaxDelay</td>
<td>ADS Ptolemy</td>
<td>2048*4</td>
</tr>
</tbody>
</table>

Notes

1. In this example, the guard interval ratio is 1/16, so the guard interval is 128.  
   (According to DVB-T, in the 2k mode the guard interval is 64, 128, 256, or 512, which corresponds to 1/32, 1/16, 1/8, or 1/4 guard interval ratio, respectively.)

2. Except for TDMA channel parameters, all parameters are related to the guard interval. If the guard interval is changed, they will be changed to the corresponding values.
3. The TDMA channel condition varies according to channel parameter modification. Parameters include Type, Pathloss, Env, Delay, and Test in the PropNADCtdma model, and Vx and Vy in the AntMobile model.

Simulation Results

Figure 8-26. ML (maximum likelihood) magnitude which shows the magnitude of ML function of the received signal used to find the FFT starting point. The maximum value appears in every 4*(2048+128) points, the maximum magnitude value is the FFT start point.

Figure 8-27. Received 64-QAM constellation after OFDM demodulation via TDMA channel interfered by adjacent band PAL signal.
Figure 8-28. Spectrum of signal received from the wireless channel

Figure 8-29. Comparison of PAL video input signal and output signal interfered by adjacent band DVB-T signal.

Benchmark
DVB-T Design Examples

- Hardware Platform: Pentium II 200 MHz, 96 MB memory
- Software Platform: Windows NT 4.0 Workstation, ADS 1.3
- Data Points: 10 OFDM symbols
- Simulation Time: 2 hours

References

DTV DVB System
OFDM 16-QAM DVB-T System without Channel Coding BER

DTV_DVBSystem_prj Design Name
DsnDTV_DVBOFDM_16QAM_BER.dsn

Features

• 16-QAM modulation and demodulation
• Guard interval
• Displays include:
  ML (maximum likelihood) estimation for OFDM frame synchronization,
  Angle (phase offset due to carrier frequency) for OFDM carrier synchronization,
  TxBit, RxBit, and Rx16QAM
• BER performance test for Gaussian, Ricean, and Rayleigh channels

Description
This OFDM adaptation for DVB-T 2k mode design example tests the BER without channel coding. Modulation mapping is 16-QAM; the guard interval ratio is 1/16. Simulation results can be compared to those of DsnDTV_DVBSystem_16QAM.dsn, which tests the BER with channel coding and interleaving.

Schematics
Figure 8-30 shows the schematic for this design. Two subnetworks are used in this design; they are shown in Figure 8-31 and Figure 8-32.
Figure 8-30. DsnDTV_DVBOFDM_16QAM_BER.dsn

Figure 8-31. sub_DVBOFDM_Mod.dsn

Figure 8-32. sub_DVBOFDM_Demod.dsn
DVB-T Design Examples

Specifications

<table>
<thead>
<tr>
<th>Symbol (Model)</th>
<th>Specification (Parameter)</th>
<th>Simulation Type</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Carriers</td>
<td>ADS Ptolemy</td>
<td></td>
<td>1705</td>
</tr>
<tr>
<td>Data</td>
<td>ADS Ptolemy</td>
<td></td>
<td>1512</td>
</tr>
<tr>
<td>IFFTSize</td>
<td>ADS Ptolemy</td>
<td></td>
<td>2048</td>
</tr>
<tr>
<td>FFTSize</td>
<td>ADS Ptolemy</td>
<td></td>
<td>2048</td>
</tr>
<tr>
<td>Guard</td>
<td>ADS Ptolemy</td>
<td></td>
<td>128</td>
</tr>
<tr>
<td>IFTOrder</td>
<td>ADS Ptolemy</td>
<td></td>
<td>11</td>
</tr>
<tr>
<td>FFTOrder</td>
<td>ADS Ptolemy</td>
<td></td>
<td>11</td>
</tr>
<tr>
<td>MaxDelay</td>
<td>ADS Ptolemy</td>
<td></td>
<td>2047</td>
</tr>
<tr>
<td>DTV_DVBChannel</td>
<td>SampleTime</td>
<td>ADS Ptolemy</td>
<td>224.0e-6/2048</td>
</tr>
</tbody>
</table>

Notes

1. In this example, the guard interval ratio is 1/16, so the guard interval is 128. (According to DVB-T, in the 2k mode the guard interval is 64, 128, 256, or 512, which corresponds to 1/32, 1/16, 1/8, or 1/4 guard interval ratio, respectively.)

2. This design can be used for three different simulations: Gaussian channel, Ricean channel (F1) and Rayleigh channel.

Simulation Results

- AWGN channel

  Additive white Gaussian noise was added to set the carrier-to-noise (CN) ratio at the input of the receiver. Figure 8-33 shows the received constellation after OFDM demodulation when the CN is 15dB and 25dB in Gaussian channel simulation. The results show the higher CN is better than the lower.
Figure 8-33. Constellation of OFDM Demodulation Signal at different CN

Figure 8-34 shows the BER at different CN.

Figure 8-34. Gaussian Channel BER
DVB-T Design Examples

- Ricean channel

Measurements of BER vs. CN were made using a Ricean channel simulator. The channel parameter is from DVB-T. In our DTV simulation system, this Ricean channel is simulated by sub_DVBChannel subnetwork, its parameters were set as follows:

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Specification</th>
<th>Simulation Type</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>sub_DVBChannel</td>
<td>NoiseGain</td>
<td>ADS Ptolemy</td>
<td>5.45^exp(CN/10.0*ln(10))</td>
</tr>
<tr>
<td>DVBChannel</td>
<td>ChannelModel</td>
<td>ADS Ptolemy</td>
<td>Fixed Reception F1</td>
</tr>
</tbody>
</table>

This simulation result was not included in this DTV package because of the limitation of DTV package size. If users want to prove the BER performance of this channel, users can set the parameters above and run this design example. The simulation time is very long.

Figure 8-35 shows the received constellation after OFDM demodulation when the CN is 15dB and 25dB, respectively in the Ricean channel simulation. The results show the result of higher CN is better than that of lower CN.

Figure 8-36 shows the BER of different CN when the simulation channel is Ricean channel.

Figure 8-35. Constellation of OFDM demodulation signal at different CN
Rayleigh channel

Measurements of BER vs. CN were made using a Rayleigh channel simulator. The channel parameter is from DVB-T. In our DTV simulation system, this Ricean channel is simulated by sub_DVBChannel subnetwork, its parameters were set as follows:

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Specification</th>
<th>Simulation Type</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>sub_DVBChannel</td>
<td>NoiseGain</td>
<td>ADS Ptolemy</td>
<td>$0.45 / \exp(\text{CN}/10.0 \ln(10))$</td>
</tr>
<tr>
<td>DVBChannel</td>
<td>ChannelModel</td>
<td>ADS Ptolemy</td>
<td>Portable Reception P1</td>
</tr>
</tbody>
</table>

This simulation result was not included in this DTV package because of the limitation of DTV package size. If users want to prove the BER performance of this channel, users can set the parameters above and run this design example. The simulation time is very long.

Figure 8-37 shows the received constellation after OFDM demodulation when the CN is 15dB and 25dB. The results show the higher CN is better than the lower.

Figure 8-38 shows the BER of different CN.
Figure 8-39 shows BERs of the three channels; the AWGN channel is the best.

Figure 8-37. Constellation of OFDM demodulation signal at different CN

Figure 8-38. Rayleigh Channel BER
Figure 8-39. Gaussian, Ricean, and Rayleigh Simulation Channel BERs

Benchmark

- Hardware Platform: Pentium II 200 MHz, 128 MB memory
- Software Platform: Windows NT 4.0 Workstation, ADS 1.3
- Data Points: 200 OFDM symbols
- Simulation Time: Approximately one hour and 38 minutes for each channel model

References

16-QAM DVB-T System Design

DTV_DVBSysstem_prj Design Name
DsnDTV_DVBSysstem_16QAM.dsn

Features
- 16-QAM modulation and demodulation
- 2k mode OFDM adaptation
- Channel coding and decoding, including Reed-Solomon, 1/2 punctured convolutional coding
- Inner and outer interleaving
- BER performance test for Gaussian, Ricean, and Rayleigh channels

Description
This DVB-T system design includes inner and outer interleaving, Reed-Solomon and convolutional coding, and full OFDM adaptation. Modulation modes and code rates are uniform 16-QAM and 1/2, respectively. In the OFDM adaptation, the full system design works in 2k mode. The IFFT/FFT is 2048; the Guard interval is 1/16 the IFFT size. Simulation channels are Gaussian, Ricean, and Rayleigh.

The byte in transmitter and receiver, the BER of the three simulation channels are shown in the simulation results. The BER of the channels in this design are compared to the DVB system without channel coding and interleaving in the DsnDTV_DVBOFDM_16QAM_BER.dsn.

Schematics
Figure 8-40 shows the schematic for this design. Four subnetworks are used in this design; they are shown in Figure 8-41 through Figure 8-44.
Figure 8-40. DsnDTV_DVBSystem_16QAM.dsn

Figure 8-41. sub_DVBOFDM_Mod.dsn

Figure 8-42. sub_DVBOFDM_Demod.dsn
DVB-T Design Examples

Figure 8-43. sub_DVBIinnerInterlv.dsn

Figure 8-44. sub_DVBIInnerDeinterlv.dsn
Specifications

<table>
<thead>
<tr>
<th>Specification (Parameter)</th>
<th>Simulation Type</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Carriers</td>
<td>ADS Ptolemy</td>
<td>1705</td>
</tr>
<tr>
<td>Data</td>
<td>ADS Ptolemy</td>
<td>1512</td>
</tr>
<tr>
<td>IFFTSize</td>
<td>ADS Ptolemy</td>
<td>2048</td>
</tr>
<tr>
<td>FFTSize</td>
<td>ADS Ptolemy</td>
<td>2048</td>
</tr>
<tr>
<td>Guard</td>
<td>ADS Ptolemy</td>
<td>128</td>
</tr>
<tr>
<td>IFFTOrder</td>
<td>ADS Ptolemy</td>
<td>11</td>
</tr>
<tr>
<td>FFTOrder</td>
<td>ADS Ptolemy</td>
<td>11</td>
</tr>
<tr>
<td>MaxDelay</td>
<td>ADS Ptolemy</td>
<td>2047</td>
</tr>
<tr>
<td>Ru</td>
<td>ADS Ptolemy</td>
<td>0.9</td>
</tr>
</tbody>
</table>

Notes

1. The propagation channel model used in this example is a standard terrestrial television propagation channel as defined in the DVB-T specification.

2. The DelayByte parameter of the punctured convolutional decoder (DTV_PuncConvDecoder model) is used to adjust delays caused by OFDM demodulation (which is one OFDM symbol) and one additional OFDM symbol delay (which corresponds to the DTV_DVBSymDeinterlv4b model) because this model uses an even number of OFDM symbols.
DVB-T Design Examples

Simulation Results

• AWGN Channel

Additive white Gaussian noise was added to set the carrier-to-noise (CN) ratio at the receiver input.

Figure 8-45 shows the transmitted and received TSP data when the CN ratio is 5dB and 8dB in Gaussian channel simulation. In Figure 8-45, the first column is the received TSP of CN = 5dB; the second column is the received TSP of CN = 8dB; the last line is the transmitted TSP. Compared to TxTSP, the first line has more errors than the second line.

<table>
<thead>
<tr>
<th>Index</th>
<th>RxTSP</th>
<th>Index</th>
<th>RxTSP</th>
</tr>
</thead>
<tbody>
<tr>
<td>1521</td>
<td>227.000</td>
<td>100.000</td>
<td>1 100.000</td>
</tr>
<tr>
<td>1522</td>
<td>224.000</td>
<td>204.000</td>
<td>2 204.000</td>
</tr>
<tr>
<td>1523</td>
<td>129.000</td>
<td>142.000</td>
<td>3 142.000</td>
</tr>
<tr>
<td>1524</td>
<td>23.000</td>
<td>23.000</td>
<td>4 23.000</td>
</tr>
<tr>
<td>1525</td>
<td>130.000</td>
<td>125.000</td>
<td>5 125.000</td>
</tr>
<tr>
<td>1526</td>
<td>129.000</td>
<td>124.000</td>
<td>6 124.000</td>
</tr>
<tr>
<td>1527</td>
<td>211.000</td>
<td>223.000</td>
<td>7 223.000</td>
</tr>
<tr>
<td>1528</td>
<td>251.000</td>
<td>251.000</td>
<td>8 251.000</td>
</tr>
<tr>
<td>1529</td>
<td>35.000</td>
<td>16.000</td>
<td>9 15.000</td>
</tr>
<tr>
<td>1530</td>
<td>34.000</td>
<td>40.000</td>
<td>10 40.000</td>
</tr>
<tr>
<td>1531</td>
<td>28.000</td>
<td>6.000</td>
<td>11 18.000</td>
</tr>
<tr>
<td>1532</td>
<td>211.000</td>
<td>211.000</td>
<td>12 211.000</td>
</tr>
<tr>
<td>1533</td>
<td>28.000</td>
<td>41.000</td>
<td>13 41.000</td>
</tr>
<tr>
<td>1534</td>
<td>251.000</td>
<td>251.000</td>
<td>14 251.000</td>
</tr>
<tr>
<td>1535</td>
<td>4.000</td>
<td>4.000</td>
<td>15 4.000</td>
</tr>
<tr>
<td>1536</td>
<td>18.000</td>
<td>135.000</td>
<td>16 155.000</td>
</tr>
<tr>
<td>1537</td>
<td>173.000</td>
<td>224.000</td>
<td>17 224.000</td>
</tr>
<tr>
<td>1538</td>
<td>210.000</td>
<td>204.000</td>
<td>18 204.000</td>
</tr>
<tr>
<td>1539</td>
<td>210.000</td>
<td>204.000</td>
<td>19 204.000</td>
</tr>
<tr>
<td>1540</td>
<td>11.000</td>
<td>11.000</td>
<td>20 11.000</td>
</tr>
<tr>
<td>1541</td>
<td>137.000</td>
<td>27.000</td>
<td>21 37.000</td>
</tr>
<tr>
<td>1542</td>
<td>231.000</td>
<td>231.000</td>
<td>22 231.000</td>
</tr>
<tr>
<td>1543</td>
<td>168.000</td>
<td>231.000</td>
<td>23 231.000</td>
</tr>
<tr>
<td>1544</td>
<td>114.000</td>
<td>114.000</td>
<td>24 114.000</td>
</tr>
<tr>
<td>1545</td>
<td>187.000</td>
<td>187.000</td>
<td>25 187.000</td>
</tr>
<tr>
<td>1546</td>
<td>129.000</td>
<td>129.000</td>
<td>26 129.000</td>
</tr>
<tr>
<td>1547</td>
<td>77.000</td>
<td>77.000</td>
<td>27 77.000</td>
</tr>
<tr>
<td>1548</td>
<td>164.000</td>
<td>164.000</td>
<td>28 164.000</td>
</tr>
<tr>
<td>1549</td>
<td>89.000</td>
<td>142.000</td>
<td>29 142.000</td>
</tr>
<tr>
<td>1550</td>
<td>189.000</td>
<td>241.000</td>
<td>30 241.000</td>
</tr>
<tr>
<td>1551</td>
<td>18.000</td>
<td>115.000</td>
<td>31 115.000</td>
</tr>
<tr>
<td>1552</td>
<td>91.000</td>
<td>91.000</td>
<td>32 91.000</td>
</tr>
<tr>
<td>1553</td>
<td>158.000</td>
<td>157.000</td>
<td>33 157.000</td>
</tr>
<tr>
<td>1554</td>
<td>189.000</td>
<td>241.000</td>
<td>34 241.000</td>
</tr>
<tr>
<td>1555</td>
<td>23.000</td>
<td>223.000</td>
<td>35 223.000</td>
</tr>
<tr>
<td>1556</td>
<td>6.000</td>
<td>6.000</td>
<td>36 6.000</td>
</tr>
<tr>
<td>1557</td>
<td>158.000</td>
<td>158.000</td>
<td>37 158.000</td>
</tr>
<tr>
<td>1558</td>
<td>158.000</td>
<td>158.000</td>
<td>38 158.000</td>
</tr>
<tr>
<td>1559</td>
<td>87.000</td>
<td>87.000</td>
<td>39 87.000</td>
</tr>
<tr>
<td>1560</td>
<td>189.000</td>
<td>211.000</td>
<td>40 211.000</td>
</tr>
</tbody>
</table>

Figure 8-45. Compare RxTSP with TxTSP with different CN

The BERs of the full system and OFDM system (without channel coding in DsnDTV_DVBOFDM_16QAM_BER.dsn) are shown in Figure 8-46. Channel coding gain is approximately 8.5dB at 0.001 BER.
According to DVB-T, in 16-QAM modulation, code rate is 1/2, and Gaussian channel model condition, the required CN for BER=0.0002 after Viterbi QEF and Reed-Solomon coding is 8.8dB. In Figure 8-46, CN is approximately 8.8dB for BER=0.0002. The simulation results shows this design is correct in the Gaussian channel.

Figure 8-46. Gaussian Channel BER
(solid line = full system; dash line = DsnDTV_DVBOFDM_16QAM_BER.dsn)
DVB-T Design Examples

- Ricean channel

Measurements of BER vs. CN were made using a Ricean channel simulator. The channel parameter is from DVB-T. In our DTV simulation system, this Ricean channel is simulated by sub_DVBChannel subnetwork, its parameters were set as follows:

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Specification</th>
<th>Simulation Type</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>sub_DVBChannel</td>
<td>NoiseGain</td>
<td>ADS Ptolemy</td>
<td>5.45/exp(CN/10.0*ln(10))</td>
</tr>
<tr>
<td>DVBChannel</td>
<td>ChannelModel</td>
<td>ADS Ptolemy</td>
<td>Fixed Reception F1</td>
</tr>
</tbody>
</table>

Figure 8-47 show the transmitted and received TSP data when CN ratio is 5dB and 8dB in the Ricean channel simulation. The first column is the received TSP of CN=5dB, the second column is the received TSP of CN=8dB; the last column is the transmitted TSP. Compared to TxTSP, the first line has more errors than the second line.

<table>
<thead>
<tr>
<th>Index</th>
<th>RxTSP</th>
<th>TxTSP</th>
</tr>
</thead>
<tbody>
<tr>
<td>2851</td>
<td>225.00</td>
<td>225.00</td>
</tr>
<tr>
<td>2852</td>
<td>264.00</td>
<td>264.00</td>
</tr>
<tr>
<td>2853</td>
<td>64.00</td>
<td>145.00</td>
</tr>
<tr>
<td>2854</td>
<td>17.00</td>
<td>22.00</td>
</tr>
<tr>
<td>2855</td>
<td>145.00</td>
<td>135.00</td>
</tr>
<tr>
<td>2856</td>
<td>180.00</td>
<td>124.00</td>
</tr>
<tr>
<td>2857</td>
<td>226.00</td>
<td>230.00</td>
</tr>
<tr>
<td>2858</td>
<td>251.00</td>
<td>251.00</td>
</tr>
<tr>
<td>2859</td>
<td>16.00</td>
<td>16.00</td>
</tr>
<tr>
<td>2860</td>
<td>116.00</td>
<td>40.00</td>
</tr>
<tr>
<td>2861</td>
<td>44.00</td>
<td>44.00</td>
</tr>
<tr>
<td>2862</td>
<td>215.00</td>
<td>211.00</td>
</tr>
<tr>
<td>2863</td>
<td>28.00</td>
<td>41.00</td>
</tr>
<tr>
<td>2864</td>
<td>251.00</td>
<td>251.00</td>
</tr>
<tr>
<td>2865</td>
<td>4.00</td>
<td>4.00</td>
</tr>
<tr>
<td>2866</td>
<td>5.00</td>
<td>151.00</td>
</tr>
<tr>
<td>2867</td>
<td>135.00</td>
<td>135.00</td>
</tr>
<tr>
<td>2868</td>
<td>230.00</td>
<td>230.00</td>
</tr>
<tr>
<td>2869</td>
<td>213.00</td>
<td>213.00</td>
</tr>
<tr>
<td>2870</td>
<td>243.00</td>
<td>243.00</td>
</tr>
<tr>
<td>2871</td>
<td>214.00</td>
<td>214.00</td>
</tr>
<tr>
<td>2872</td>
<td>137.00</td>
<td>137.00</td>
</tr>
<tr>
<td>2873</td>
<td>251.00</td>
<td>251.00</td>
</tr>
<tr>
<td>2874</td>
<td>230.00</td>
<td>230.00</td>
</tr>
<tr>
<td>2875</td>
<td>243.00</td>
<td>243.00</td>
</tr>
<tr>
<td>2876</td>
<td>117.00</td>
<td>117.00</td>
</tr>
<tr>
<td>2877</td>
<td>27.00</td>
<td>27.00</td>
</tr>
<tr>
<td>2878</td>
<td>164.00</td>
<td>164.00</td>
</tr>
<tr>
<td>2879</td>
<td>142.00</td>
<td>142.00</td>
</tr>
<tr>
<td>2880</td>
<td>243.00</td>
<td>243.00</td>
</tr>
<tr>
<td>2881</td>
<td>161.00</td>
<td>161.00</td>
</tr>
<tr>
<td>2882</td>
<td>91.00</td>
<td>91.00</td>
</tr>
<tr>
<td>2883</td>
<td>158.00</td>
<td>158.00</td>
</tr>
<tr>
<td>2884</td>
<td>175.00</td>
<td>175.00</td>
</tr>
<tr>
<td>2885</td>
<td>227.00</td>
<td>227.00</td>
</tr>
<tr>
<td>2886</td>
<td>88.00</td>
<td>88.00</td>
</tr>
<tr>
<td>2887</td>
<td>123.00</td>
<td>123.00</td>
</tr>
<tr>
<td>2888</td>
<td>152.00</td>
<td>152.00</td>
</tr>
<tr>
<td>2889</td>
<td>215.00</td>
<td>215.00</td>
</tr>
<tr>
<td>2890</td>
<td>211.00</td>
<td>211.00</td>
</tr>
</tbody>
</table>

Figure 8-47. Compare RxTSP with TxTSP with different CN
The BERs of the full system and OFDM system (without channel coding in DsnDTV_DVBOFDM_16QAM_BER.dsn) are shown in Figure 8-48. Channel coding gain is approximately 10dB at 0.001 BER. According to DVB-T, in 16-QAM modulation, code rate is 1/2, and Gaussian channel model condition, the required CN for BER=0.0002 after Viterbi QEF after Reed-Solomon coding is 9.6dB. In Figure 8-48, CN is approximately 9.6dB for BER=0.0002. The simulation results shows this design is correct in the Ricean channel.
• Rayleigh channel

Measurements of BER vs. CN were made using a Rayleigh channel simulator. The channel parameter is from DVB-T. In our DTV simulation system, this Ricean channel is simulated by sub_DVBChannel subnetwork, its parameters were set as follows:

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Specification</th>
<th>Simulation Type</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>sub_DVBChannel</td>
<td>NoiseGain</td>
<td>ADS Ptolemy</td>
<td>0.45/exp(CN/10.0*ln(10))</td>
</tr>
<tr>
<td>DVBChannel</td>
<td>ChannelModel</td>
<td>ADS Ptolemy</td>
<td>Portable Reception P1</td>
</tr>
</tbody>
</table>

This simulation result was not included in this DTV package because of the limitation of DTV package size. To prove the BER performance of this channel, set the parameters and run this design example; simulation time is very long.

Figure 8-49 shows the transmitted and received TSP data when CN is 5dB and 8dB in Rayleigh channel simulation. The first column is the received TSP of CN=5dB, the second column is the received TSP of CN=8dB; the last column is the transmitted TSP. Compared to the TxTSP, the first column shows many errors, while the second column has fewer errors.
Figure 8-49. Compare RxTSP with TxTSP with different CN

The BERs of the full system and OFDM system (without channel coding in DsnDTV_DVBOFDM_16QAM_BER.dsn) are shown in Figure 8-50. Channel coding gain is approximately 12.0dB at 0.001 BER. According to the DVB-T, in 16-QAM modulation, code rate is 1/2, and Rayleigh channel model condition, the required CN for BER=0.0002 after Viterbi QEF after Reed-Solomon coding is 11.2dB. In Figure 8-50, CN is approximately 13.8dB for BER=0.0002.
The simulated system performance in DVB-T Appendix A is obtained at Perfect channel estimation and without phase noise condition; the performance in DVB-T Table A.1 is the best performance. In our simulation, channel estimation is not perfect because of noise (the channel estimation may have some distortion). Moreover, the frequency equalizer may enhance noise power because of fading. Therefore, the 2.6dB difference between our simulation and DVB-T is suitable because there is no fading in Gaussian and Ricean channels.

Figure 8-51 is shown the BERs of all the three simulation channels. From Figure 8-51, the performance of the Gaussian channel is the best, then the Ricean channel, the performance of the Rayleigh channel is the worst.
Figure 8-51. Gaussian, Ricean, Rayleigh Simulation Channel BERs

Benchmark

- Hardware Platform: Pentium II 200 MHz, 128 MB memory
- Software Platform: Windows NT 4.0 Workstation, ADS 1.3
- Data Points: 188*3 Bytes
- Simulation Time: Approximately 16 hours for Rayleigh channel model, 10 hours for Ricean channel model, 9 hours for Gaussian channel model

References

Hierarchical 64-QAM DVB-T System Design

DTV_DVBSYSTEM_prj Design Name
DsnDTV_DVBSYSTEM_Hier64QAM.dsn

Features

• Channel coding and decoding, including Reed-Solomon coding, punctured convolutional coding
• 2k mode OFDM adaptation
• Inner and outer interleaving
• Hierarchical 64-QAM modulation and demodulation

Description
This is a hierarchical DVB-T system design example. It includes inner and outer interleaving, Reed-Solomon and convolutional coding and full OFDM adaptation system. The modulation mode is hierarchical 64-QAM. Code rates for HP level(r1) and LP level(r2) are 1/2 and 5/6, respectively. In the OFDM adaptation, the full system design works in 8k mode. The IFFT/FFT size is 8192 and the Guard interval is 1/16 of IFFT size. The simulation channel is the DVB channel. The byte in transmitter and receiver are shown in the simulation results.

Schematics
Figure 8-52 shows the schematic for this design. Four subnetworks are used in this design; they are shown in Figure 8-53 through Figure 8-56.
Figure 8-52. DsnDTV_DVBSystem_Hier64QAM.dsn

Figure 8-53. sub_DVBOFDM_Mod.dsn

Figure 8-54. sub_DVBOFDM_Demod.dsn
DVB-T Design Examples

Figure 8-55. sub_DVBHier_InnerInterlvdsn

Figure 8-56. sub_DVBHier_InnerDeinterlvdsn
Specifications

<table>
<thead>
<tr>
<th>Specification (Parameter)</th>
<th>Simulation Type</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Carriers</td>
<td>ADS Ptolemy</td>
<td>6817</td>
</tr>
<tr>
<td>Data</td>
<td>ADS Ptolemy</td>
<td>6048</td>
</tr>
<tr>
<td>IFFTSize</td>
<td>ADS Ptolemy</td>
<td>8192</td>
</tr>
<tr>
<td>FFTSize</td>
<td>ADS Ptolemy</td>
<td>8192</td>
</tr>
<tr>
<td>Guard</td>
<td>ADS Ptolemy</td>
<td>512</td>
</tr>
<tr>
<td>IFFTOrder</td>
<td>ADS Ptolemy</td>
<td>13</td>
</tr>
<tr>
<td>FFTOrder</td>
<td>ADS Ptolemy</td>
<td>13</td>
</tr>
<tr>
<td>MaxDelay</td>
<td>ADS Ptolemy</td>
<td>8192</td>
</tr>
<tr>
<td>Ru</td>
<td>ADS Ptolemy</td>
<td>0.9</td>
</tr>
</tbody>
</table>

Notes

1. The propagation channel model used in this example is a standard terrestrial television propagation channel which is defined in the DVB-T specification.

2. The DelayByte parameter of the punctured convolutional decoder (DTV_PuncConvDecoder model) is used to adjust delays caused by OFDM demodulation (which is one OFDM symbol) and one additional OFDM symbol delay (which corresponds to the DTV_DVB_SYMDeinterlv6b model) because this model uses an even number of OFDM symbols.

Simulation Results

Figure 8-57 shows the transmitted HP (high priority) TSP data and the received HP TSP data. Figure 8-58 shows the transmitted LP (low priority) TSP data and the received LP low data.
### DVB-T Design Examples

**Figure 8-57. Compare HP RxTSP with HP TxTSP**

<table>
<thead>
<tr>
<th>Index</th>
<th>RxTSP</th>
<th>Index</th>
<th>TxTSP</th>
</tr>
</thead>
<tbody>
<tr>
<td>6766</td>
<td>100 000</td>
<td>0</td>
<td>100 000</td>
</tr>
<tr>
<td>6768</td>
<td>203 000</td>
<td>1</td>
<td>203 000</td>
</tr>
<tr>
<td>6787</td>
<td>145 000</td>
<td>2</td>
<td>145 000</td>
</tr>
<tr>
<td>6771</td>
<td>23 000</td>
<td>3</td>
<td>23 000</td>
</tr>
<tr>
<td>6772</td>
<td>136 000</td>
<td>4</td>
<td>136 000</td>
</tr>
<tr>
<td>6773</td>
<td>124 000</td>
<td>5</td>
<td>124 000</td>
</tr>
<tr>
<td>6774</td>
<td>223 000</td>
<td>6</td>
<td>223 000</td>
</tr>
<tr>
<td>6775</td>
<td>251 000</td>
<td>7</td>
<td>251 000</td>
</tr>
<tr>
<td>6776</td>
<td>18 000</td>
<td>8</td>
<td>18 000</td>
</tr>
<tr>
<td>6777</td>
<td>48 000</td>
<td>9</td>
<td>48 000</td>
</tr>
<tr>
<td>6778</td>
<td>5 000</td>
<td>10</td>
<td>5 000</td>
</tr>
<tr>
<td>6779</td>
<td>101 000</td>
<td>11</td>
<td>101 000</td>
</tr>
<tr>
<td>6780</td>
<td>241 000</td>
<td>12</td>
<td>241 000</td>
</tr>
<tr>
<td>6781</td>
<td>251 000</td>
<td>13</td>
<td>251 000</td>
</tr>
<tr>
<td>6782</td>
<td>4 000</td>
<td>14</td>
<td>4 000</td>
</tr>
<tr>
<td>6783</td>
<td>153 000</td>
<td>15</td>
<td>153 000</td>
</tr>
<tr>
<td>6784</td>
<td>234 000</td>
<td>16</td>
<td>234 000</td>
</tr>
<tr>
<td>6785</td>
<td>204 000</td>
<td>17</td>
<td>204 000</td>
</tr>
<tr>
<td>6786</td>
<td>243 000</td>
<td>18</td>
<td>243 000</td>
</tr>
<tr>
<td>6787</td>
<td>15 000</td>
<td>19</td>
<td>15 000</td>
</tr>
<tr>
<td>6788</td>
<td>37 000</td>
<td>20</td>
<td>37 000</td>
</tr>
<tr>
<td>6789</td>
<td>224 000</td>
<td>21</td>
<td>224 000</td>
</tr>
<tr>
<td>6790</td>
<td>231 000</td>
<td>22</td>
<td>231 000</td>
</tr>
<tr>
<td>6791</td>
<td>114 000</td>
<td>23</td>
<td>114 000</td>
</tr>
<tr>
<td>6792</td>
<td>187 000</td>
<td>24</td>
<td>187 000</td>
</tr>
<tr>
<td>6793</td>
<td>126 000</td>
<td>25</td>
<td>126 000</td>
</tr>
<tr>
<td>6794</td>
<td>71 000</td>
<td>26</td>
<td>71 000</td>
</tr>
<tr>
<td>6795</td>
<td>164 000</td>
<td>27</td>
<td>164 000</td>
</tr>
<tr>
<td>6796</td>
<td>142 000</td>
<td>28</td>
<td>142 000</td>
</tr>
<tr>
<td>6797</td>
<td>244 000</td>
<td>29</td>
<td>244 000</td>
</tr>
<tr>
<td>6798</td>
<td>116 000</td>
<td>30</td>
<td>116 000</td>
</tr>
<tr>
<td>6799</td>
<td>91 000</td>
<td>31</td>
<td>91 000</td>
</tr>
<tr>
<td>6800</td>
<td>169 000</td>
<td>32</td>
<td>169 000</td>
</tr>
<tr>
<td>6801</td>
<td>47 000</td>
<td>33</td>
<td>47 000</td>
</tr>
<tr>
<td>6802</td>
<td>223 000</td>
<td>34</td>
<td>223 000</td>
</tr>
<tr>
<td>6803</td>
<td>9 000</td>
<td>35</td>
<td>9 000</td>
</tr>
<tr>
<td>6804</td>
<td>159 000</td>
<td>36</td>
<td>159 000</td>
</tr>
<tr>
<td>6805</td>
<td>154 000</td>
<td>37</td>
<td>154 000</td>
</tr>
</tbody>
</table>
Figure 8-58. Compare LP RxTSP with LP TxTSP

Benchmark

- Hardware Platform: Pentium II 200 MHz, 128 MB memory
- Software Platform: Windows NT 4.0 Workstation, ADS 1.3
- Data Points: 188*3 Bytes
- Simulation Time: Approximately 1.5 hours

References

DVB-T Design Examples
Chapter 9: ISDB-T Design Examples

Introduction

DTV example designs can be accessed from the /examples/dtv directory. Schematics and simulation results for the following design examples are described in this chapter.

DTV_ISDBOFDM_prj
- DsnDTV_ISDBOFDM_64QAM.dsn
- DsnDTV_ISDBOFDM_DQPSK.dsn
- DsnDTV_ISDBOFDM_NTSCInterference.dsn
- DsnDTV_ISDBOFDM_TwoLay.dsn
- DsnDTV_ISDBOFDM_ThrLay.dsn

DTV_ISDBSystem_prj
- DsnDTV_ISDBOFDM_64QAM_BER.dsn
- DsnDTV_ISDBOFDM_DQPSK_BER.dsn
- DsnDTV_ISDBOneLay_64QAM.dsn
- DsnDTV_ISDBOneLay_DQPSK.dsn
- DsnDTV_ISDBTwoLay_System.dsn
- DsnDTV_ISDBThrLay_System.dsn
- DsnDTV_TMCCMod.dsn
- DsnDTV_TMCCThrLay.dsn
ISDB-T Design Examples

**DTV ISDB OFDM**

**OFDM 64-QAM Modulation and Demodulation in ISDB-T Systems**

DTV_ISDBOFDM_prj Design Name  
DsnDTV_ISDBOFDM_64QAM.dsn

**Features**

- 64-QAM modulation and demodulation
- Guard interval
- TDMA multipath fading channel with Doppler shift. The type of multipath can be selected; Doppler shift can be determined by the mobile speed setting.
- Displays include:
  - ML (maximum likelihood)
  - RxSpectrum (spectrum of received signal from the channel)
  - RxQAM (received signal constellation after OFDM demodulation)

**Description**

This design is an example of modulation, transmission and demodulation sections via a TDMA multipath channel with Doppler shift. In this example, a single layer transmission is used in mode 3, which includes 13 segments. Modulation mapping is 64-QAM; the guard interval ratio is 1/16.

After 64-QAM data mapping, the 13 OFDM segments are formed by adding TMCC and AC data, and scattered pilots. After the spectral order of the segments is changed, 8192-point inverse FFT (IFFT) is performed. After inserting the guard interval, the complex signal is transmitted; the transmitted signal length is 8192+guard interval.

In the receiver, the FFT starting point is determined by the ML function of the received signal. After FFT, the transmitted signal is recovered by a simple frequency equalizer. According to the ISDB-T, the mapping data was received. Data is recovered by the 64-QAM demodulator.

System performance can be viewed during simulation.

**Schematics**
Figure 9-1 shows the schematic for this design; subnetwork designs are shown in Figure 9-2 and Figure 9-3.
ISDB-T Design Examples

Specifications

<table>
<thead>
<tr>
<th>Symbol (Model)</th>
<th>Specification (Parameter)</th>
<th>Simulation Type</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>PropNADCtdma</td>
<td>Type</td>
<td>ADS Ptolemy</td>
<td>TwoPath</td>
</tr>
<tr>
<td>PropNADCtdma</td>
<td>Pathloss</td>
<td>ADS Ptolemy</td>
<td>No</td>
</tr>
<tr>
<td>PropNADCtdma</td>
<td>Env</td>
<td>ADS Ptolemy</td>
<td>TypicalUrban</td>
</tr>
<tr>
<td>PropNADCtdma</td>
<td>Delay</td>
<td>ADS Ptolemy</td>
<td>1.0 μsec</td>
</tr>
<tr>
<td>PropNADCtdma</td>
<td>Test</td>
<td>ADS Ptolemy</td>
<td>Tap1</td>
</tr>
<tr>
<td>AntMobile</td>
<td>Vx</td>
<td>ADS Ptolemy</td>
<td>30 km/hr</td>
</tr>
<tr>
<td>AntMobile</td>
<td>Vy</td>
<td>ADS Ptolemy</td>
<td>0.0 km/hr</td>
</tr>
</tbody>
</table>

Notes

1. In this example, the guard interval ratio is 1/16, so the guard interval is 512. (According to ISDB-T, in mode 3 the guard interval is 256, 512, 1024, or 2048, which corresponds to 1/32, 1/16, 1/8, or 1/4 guard interval ratio, respectively.)

2. All parameters are related to the guard interval (except TDMA channel parameters). If the guard interval is changed, parameters will be changed to the corresponding values.

3. The TDMA channel condition varies according to channel parameter modification. Parameters include Type, Pathloss, Env, Delay, and Test in the PropNADCtdma model, and Vx and Vy in the AntMobile model.

Simulation Results

Figure 9-4. ML magnitude, which shows the magnitude of ML function of received signal used to find the FFT start. The maximum value appears in every (8192+512) points, the point of the maximum magnitude value is the FFT start.
Figure 9-5. Received constellation after OFDM demodulation by TDMA channel. Results show OFDM system performance.

Figure 9-6. Spectrum of signal received from wireless channel. Center frequency is 611MHz

Figure 9-7. Adjacent-Channel Power Ratio
The equation is:
\[
ACPR = acpr_vr(DsnDTV_ISDBOFDM_64QAM,RxSignal,50.0,\{-2.808MHz,2.808MHz\},\{-8.808MHz,-3.192MHz\},\{3.192MHz,8.808MHz\})
\]
ISDB-T Design Examples

Figure 9-8. Relative magnitude error, which is defined as:
\[ EVM = \frac{\text{mag}(\text{DsnDTV}_\text{ISDBOFDM}_\text{64QAM}.. \text{RxQAM}-\text{DsnDTV}_\text{ISDBOFDM}_\text{64QAM}.. \text{TxQAM}))}{\text{mean}(\text{mag}(\text{DsnDTV}_\text{ISDBOFDM}_\text{64QAM}.. \text{TxQAM}))} \]

Figure 9-9. EVM histogram, which shows most EVM < 0.1. EVMhist is defined as:
\[ \text{EVMhist} = \text{histogram}(EVM, 1001, 0.0, 1.0) \]

| EVMavr | 0.029 |

Figure 9-10. EVMavr = mean(EVM)

**Benchmark**

- Hardware Platform: Pentium II 200 MHz, 96 MB memory

---

9-6 DTV ISDB OFDM
• Software Platform: WindowsNT 4.0 Workstation, ADS 1.3
• Data Points: 10 symbols
• Simulation Time: 78 seconds

References

OFDM DQPSK Modulation and Demodulation in ISDB-T Systems

DTV_ISDBOFDM_prj Design Name
DsnDTV_ISDBOFDM_DQPSK.dsn

Features

• DQPSK modulation and demodulation
• Guard interval
• TDMA multipath fading channel with Doppler shift. The multipath type can be selected and the Doppler shift can be determined by setting the mobile’s speed.
• Displays include:
  ML (maximum likelihood function)
  RxSpectrum (spectrum of signal received from the channel)
  RxDQPSK (received signal constellation after OFDM demodulation)

Description

This design is an example of modulation, transmission and demodulation sections via a TDMA multipath channel with Doppler shift. In this example, a single layer transmission is used in mode 3, which includes 13 segments. Modulation mapping is DQPSK; the guard interval ratio is 1/16.

After DQPSK data mapping, the 13 OFDM segments are formed by adding TMCC and AC data, and scattered pilots. After the spectral order of the segments is changed, 8192-point inverse FFT (IFFT) is performed. After inserting the guard interval, the complex signal is transmitted; the transmitted signal length is 8192+guard interval.

In the receiver, the FFT starting point is determined by the ML function of the received signal. After FFT, the transmitted signal is recovered by simple frequency equalizer. According to the ISDB-T, the mapping data was received. TSP data is recovered by the DQPSK demodulator.

System performance can be viewed during simulation.

Schematics

Figure 9-11 shows the schematic for this design; subnetwork designs are shown in Figure 9-12 and Figure 9-13.
Figure 9-11. DsnDTV_ISDBOFDM_DQPSK.dsn

Figure 9-12. sub_ISDBOFDM_Mod.dsn

Figure 9-13. sub_ISDBOFDM_Demod.dsn
ISDB-T Design Examples

Specifications

<table>
<thead>
<tr>
<th>Symbol (Model)</th>
<th>Specification (Parameter)</th>
<th>Simulation Type</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>PropNADCtdma</td>
<td>Type</td>
<td>ADS Ptolemy</td>
<td>TwoPath</td>
</tr>
<tr>
<td>PropNADCtdma</td>
<td>Pathloss</td>
<td>ADS Ptolemy</td>
<td>No</td>
</tr>
<tr>
<td>PropNADCtdma</td>
<td>Env</td>
<td>ADS Ptolemy</td>
<td>TypicalUrban</td>
</tr>
<tr>
<td>PropNADCtdma</td>
<td>Delay</td>
<td>ADS Ptolemy</td>
<td>6 µsec</td>
</tr>
<tr>
<td>PropNADCtdma</td>
<td>Test</td>
<td>ADS Ptolemy</td>
<td>Tap1</td>
</tr>
<tr>
<td>AntMobile</td>
<td>Vx</td>
<td>ADS Ptolemy</td>
<td>30 km/hr</td>
</tr>
<tr>
<td>AntMobile</td>
<td>Vy</td>
<td>ADS Ptolemy</td>
<td>0.0 km/hr</td>
</tr>
</tbody>
</table>

Notes

1. In this example, the guard interval ratio is 1/16, so the guard interval is 512. (According to ISDB-T, in mode 3 the guard interval is 256, 512, 1024, or 2048, which corresponds to 1/32, 1/16, 1/8, or 1/4 guard interval ratio, respectively.)

2. All parameters are related to the guard interval (except TDMA channel parameters). If the guard interval is changed, parameters will be changed to the corresponding values.

3. The TDMA channel condition varies according to channel parameter modification. Parameters include Type, Pathloss, Env, Delay, and Test in the PropNADCtdma model, and Vx and Vy in the AntMobile model.

Simulation Results

![Figure 9-14. ML magnitude which shows the magnitude of ML function of the received signal used to find the FFT start. The maximum value appears in every (8192+512) points, the point of the maximum magnitude value is the FFT start.](image)
Figure 9-15. Received constellation after OFDM demodulation by TDMA channel. Results show OFDM system performance.

Figure 9-16. Spectrum of Signal Received from Wireless Channel. Center frequency is 611 MHz

<table>
<thead>
<tr>
<th>ACPR</th>
<th>ACPR1</th>
<th>ACPR2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>-50.091</td>
<td>-50.360</td>
</tr>
</tbody>
</table>

Figure 9-17. Adjacent-Channel Power Ratio
The equation is:

\[
ACPR = acpr_vr(DsnDTV_ISDBOFDM_DQPSK..RxSignal,50.0,\{2.808MHz,2.808MHz\},\{-8.808MHz,-3.192MHz\},\{3.192MHz,8.808MHz\})
\]
Figure 9-18. Relative magnitude error, which is defined as:
\[ \text{EVM} = \frac{\text{mag}(\text{DsnDTV\_ISDBOFDM\_DQPSK}\_\text{RxDQPSK} - \text{DsnDTV\_ISDBOFDM\_DQPSK}\_\text{TxDQPSK})}{\text{mean}(\text{mag}(\text{DsnDTV\_ISDBOFDM\_DQPSK}\_\text{TxDQPSK}))} \]

Figure 9-19. Histogram of EVM, which shows most EVM < 0.1.
\[ \text{EVMhist} = \text{histogram}(\text{EVM}, 1001, 0.0, 1.0) \]

**Benchmark**
- Hardware Platform: Pentium II 200 MHz, 96 MB memory
- Software Platform: WindowsNT 4.0 Workstation, ADS 1.3
- Data Points: 10 symbols
- Simulation Time: 110 seconds
References

DTV and NTSC Signal Interference Test in ISDB-T Systems

DTV_ISDBOFDM_prj Design Name
DsnDTV_ISDBOFDM_NTSCInterference.dsn

Features
- 64-QAM modulation and demodulation
- Adjacent channel interference between NTSC and ISDB-T signal
- Displays include:
  - ML (maximum likelihood function),
  - RxSpectrum (spectrum of signal received from the channel),
  - RxQAM (received signal constellation after OFDM demodulation),
  - VideoIn, VideoOut
- Comparison of analog NTSC TV signal and received NTSC TV signal that has adjacent channel ISDB-T signal interference

Description
This design is an OFDM example that tests the adjacent interference between the analog NTSC TV signal and the digital ISDB-T signal. ISDB-T signal experiences weak interference while the analog NTSC signal experiences strong interference. This effect is saved in Data Display (DsnDTV_ISDBOFDM_NTSCInterference1.dds). The analog NTSC spectrum experiences interference at frequencies over 6 MHz. The ISDB-T signal center frequency is 611.15 MHz, which is shifted 150 kHz.

Schematics
Figure 9-20 shows the schematic for this design; subnetwork designs are shown in Figure 9-21 through Figure 9-24.
Notes

1. In this example, the guard interval ratio is 1/16, so the guard interval is 512. (According to ISDB-T, in mode 3 the guard interval is 256, 512, 1024, or 2048, which corresponds to 1/32, 1/16, 1/8, or 1/4 guard interval ratio, respectively.)

2. All parameters are related to the guard interval (except TDMA channel parameters). If the guard interval is changed, parameters will be changed to the corresponding values.

3. The TDMA channel condition varies according to channel parameter modification. Parameters include Type, Pathloss, Env, Delay, and Test in the PropNADCtdma model, and Vx and Vy in the AntMobile model.
Simulation Results

Figure 9-25. ML magnitude which shows the magnitude of ML function of received signal used to find FFT start. Maximum value appears in every 4*(8192+512) points, the point of the maximum magnitude value is the FFT start.

Figure 9-26. Received 64QAM Constellation after OFDM demodulation by TDMA channel with NTSC Signal Interference.
ISDB-T Design Examples

Figure 9-27. Spectrum of Signal Received from Wireless Channel

Figure 9-28. NTSC Video Input and Output Signals with ISDB-T Signal Interference
Benchmark

- Hardware Platform: Pentium II 200 MHz, 96 MB memory
- Software Platform: WindowsNT 4.0 Workstation, ADS 1.3
- Data Points: 10 symbols
- Simulation Time: 8.5 hours

References

OFDM 2-Layer Modulation and Demodulation in ISDB-T Systems

DTV_ISDBOFDM_prj Design Name
DsnDTV_ISDBOFDM_TwoLay.dsn

Features
- 64-QAM modulation and demodulation
- DQPSK modulation and demodulation
- Guard interval
- TDMA multipath fading channel with Doppler shift. The kind of the multipath can be selected and the Doppler shift can be determined by setting the mobile's speed.
- Displays include:
  - ML (maximum likelihood function)
  - RxSpectrum (spectrum of signal received from the channel)
  - RxQAM (received signal constellation after OFDM demodulation)
  - RxDQPSK

Description
This design is a 2-layer OFDM design example of ISDB-T. It includes 13 segments (5 segments for Layer A, 8 segments for Layer B). Modulation modes are DQPSK and 64-QAM in Layer A and Layer B, respectively.

The IFFT/FFT size is 2048; the guard interval is 1/8 of IFFT size. The simulation channel is the TDMA channel. The byte in transmitter and receiver, the received DQPSK constellation and 64-QAM constellation are shown in the simulation results.

Schematics
Figure 9-29 shows the schematic for this design; subnetwork designs are shown in Figure 9-30 and Figure 9-31.
ISDB-T Design Examples

Specifications

<table>
<thead>
<tr>
<th>Symbol (Model)</th>
<th>Specification (Parameter)</th>
<th>Simulation Type</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Carriers</td>
<td>ADS Ptolemy</td>
<td></td>
<td>108</td>
</tr>
<tr>
<td>Segments</td>
<td>ADS Ptolemy</td>
<td></td>
<td>13</td>
</tr>
<tr>
<td>IFFTSize</td>
<td>ADS Ptolemy</td>
<td></td>
<td>2048</td>
</tr>
<tr>
<td>FFTSize</td>
<td>ADS Ptolemy</td>
<td></td>
<td>2048</td>
</tr>
<tr>
<td>Guard</td>
<td>ADS Ptolemy</td>
<td></td>
<td>256</td>
</tr>
<tr>
<td>IFFTOrder</td>
<td>ADS Ptolemy</td>
<td></td>
<td>11</td>
</tr>
<tr>
<td>FFTOrder</td>
<td>ADS Ptolemy</td>
<td></td>
<td>11</td>
</tr>
<tr>
<td>MaxDelay</td>
<td>ADS Ptolemy</td>
<td></td>
<td>2047</td>
</tr>
<tr>
<td>SegmentsA</td>
<td>ADS Ptolemy</td>
<td></td>
<td>5</td>
</tr>
<tr>
<td>SegmentsB</td>
<td>ADS Ptolemy</td>
<td></td>
<td>8</td>
</tr>
<tr>
<td>Ru</td>
<td>ADS Ptolemy</td>
<td></td>
<td>0.9</td>
</tr>
<tr>
<td>PropNADCtdma</td>
<td>Type</td>
<td>ADS Ptolemy</td>
<td>FlatFading</td>
</tr>
<tr>
<td>PropNADCtdma</td>
<td>Pathloss</td>
<td>ADS Ptolemy</td>
<td>No</td>
</tr>
<tr>
<td>PropNADCtdma</td>
<td>Env</td>
<td>ADS Ptolemy</td>
<td>TypicalUrban</td>
</tr>
<tr>
<td>PropNADCtdma</td>
<td>Test</td>
<td>ADS Ptolemy</td>
<td>Tap1</td>
</tr>
<tr>
<td>AntMobile</td>
<td>Vx</td>
<td>ADS Ptolemy</td>
<td>30 km/hr</td>
</tr>
<tr>
<td>AntMobile</td>
<td>Vy</td>
<td>ADS Ptolemy</td>
<td>0.0 km/hr</td>
</tr>
</tbody>
</table>

Notes

1. The modification of the guard interval is according to the ISDB-T. In mode 1, the value of the guard interval is 64, 128, 256, and 512 corresponding to the 1/32, 1/16, 1/8, and 1/4 guard interval ratio, respectively. In this example, the guard interval ratio is 1/8, so the guard interval is 256.

2. All parameters are related to the guard interval (except TDMA channel parameters). If the guard interval is changed, parameters will be changed to the corresponding values.

3. The TDMA channel condition varies according to channel parameter modification. Parameters include Type, Pathloss, Env, Delay, and Test in the PropNADCtdma model, and Vx and Vy in the AntMobile model.
Simulation Results

Figure 9-32. ML magnitude which shows the magnitude of ML function of the received signal used to find the FFT start. The maximum value appears in every (2048+128) points, the point of the maximum magnitude value is the FFT start.

Figure 9-33. Received 64-QAM constellation after OFDM demodulation by TDMA channel. Results show OFDM system performance.

Figure 9-34. Received DQPSK constellation after OFDM demodulation by TDMA channel. Results show OFDM system performance.
ISDB-T Design Examples

Figure 9-35. Spectrum of Signal Received from Wireless Channel.
Center frequency is 611 MHz

Figure 9-36. Adjacent-Channel Power Ratio
The equation is:
\[
ACPR = \text{acpr}_v(DsnDTV_ISDBOFDM_TwoLay.RxSignal,50.0, \\
\{2.808MHz,2.808MHz\}, \{8.808MHz,-3.192MHz\}, \{3.192MHz,8.808MHz\})
\]

Benchmark
- Hardware Platform: Pentium II 200 MHz, 96 MB memory
- Software Platform: WindowsNT 4.0 Workstation, ADS 1.3
- Data Points: 10 symbols
- Simulation Time: 20 seconds

References
OFDM 3-Layer Modulation and Demodulation in ISDB-T Systems

DTV_ISDBOFDM_prj Design Name
DsnDTV_ISDBOFDM_ThrLay.dsn

Features

- 64-QAM modulation and demodulation
- DQPSK modulation and demodulation
- 16-QAM modulation and demodulation
- Guard interval
- TDMA multipath fading channel with Doppler shift. The kind of the multipath can be selected and the Doppler shift can be determined by setting the mobile's speed.
- Displays include:
  - ML (maximum likelihood function)
  - RxSpectrum (spectrum of signal received from the channel)
  - RxQAM (received signal constellation after OFDM demodulation)
  - Rx16QAM
  - RxDQPSK

Description

This design is a 3-layer OFDM design example of ISDB-T. It includes 13 segments (1 segment for Layer A, 5 segments for Layer B, and 7 segments for Layer C). The modulation modes are 64-QAM, and DQPSK, and 16-QAM, in Layer A, Layer B, and Layer C, respectively. In the OFDM adaptation, the 3-layer OFDM design works in mode 1. The IFFT/FFT size is 2048; the Guard interval is 1/16 of IFFT size. The simulation channel is the TDMA channel.

The received 64-QAM, DQPSK, and 16-QAM constellations are shown in the simulation results.

Schematics

Figure 9-37 shows the schematic for this design; subnetwork designs are shown in Figure 9-38 and Figure 9-39.
ISDB-T Design Examples

Figure 9-37. DsnDTV_ISDBOFDM_ThrLay.dsn

Figure 9-38. sub_ISDBOFDM_ThrLayMod.dsn
Table 9-39. sub-ISDBOFDM_ThrLayDemod.dsn Specifications

<table>
<thead>
<tr>
<th>Symbol (Model)</th>
<th>Specification (Parameter)</th>
<th>Simulation Type</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Carriers</td>
<td>ADS Ptolemy</td>
<td>108</td>
</tr>
<tr>
<td></td>
<td>Segments</td>
<td>ADS Ptolemy</td>
<td>13</td>
</tr>
<tr>
<td></td>
<td>IFFTSize</td>
<td>ADS Ptolemy</td>
<td>2048</td>
</tr>
<tr>
<td></td>
<td>FFTSize</td>
<td>ADS Ptolemy</td>
<td>2048</td>
</tr>
<tr>
<td></td>
<td>Guard</td>
<td>ADS Ptolemy</td>
<td>128</td>
</tr>
<tr>
<td></td>
<td>IFFTOrder</td>
<td>ADS Ptolemy</td>
<td>11</td>
</tr>
<tr>
<td></td>
<td>FFTOrder</td>
<td>ADS Ptolemy</td>
<td>11</td>
</tr>
<tr>
<td></td>
<td>MaxDelay</td>
<td>ADS Ptolemy</td>
<td>2047</td>
</tr>
<tr>
<td></td>
<td>SegmentsA</td>
<td>ADS Ptolemy</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>SegmentsB</td>
<td>ADS Ptolemy</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>SegmentsC</td>
<td>ADS Ptolemy</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td>Ru</td>
<td>ADS Ptolemy</td>
<td>0.9</td>
</tr>
<tr>
<td>PropNADCtdma</td>
<td>Type</td>
<td>ADS Ptolemy</td>
<td>TwoPath</td>
</tr>
<tr>
<td>PropNADCtdma</td>
<td>Pathloss</td>
<td>ADS Ptolemy</td>
<td>No</td>
</tr>
<tr>
<td>PropNADCtdma</td>
<td>Env</td>
<td>ADS Ptolemy</td>
<td>TypicalUrban</td>
</tr>
<tr>
<td>PropNADCtdma</td>
<td>Delay</td>
<td>ADS Ptolemy</td>
<td>0.5 µsec</td>
</tr>
<tr>
<td>PropNADCtdma</td>
<td>Test</td>
<td>ADS Ptolemy</td>
<td>Tap1</td>
</tr>
</tbody>
</table>

Figure 9-39. sub-ISDBOFDM_ThrLayDemod.dsn
ISDB-T Design Examples

<table>
<thead>
<tr>
<th>Symbol (Model)</th>
<th>Specification (Parameter)</th>
<th>Simulation Type</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>AntMobile</td>
<td>Vx</td>
<td>ADS Ptolemy</td>
<td>30 km/hr</td>
</tr>
<tr>
<td>AntMobile</td>
<td>Vy</td>
<td>ADS Ptolemy</td>
<td>0.0 km/hr</td>
</tr>
</tbody>
</table>

Notes

1. The modification of the guard interval is according to the ISDB-T. In mode 1, the value of the guard interval is 64, 128, 256, and 512 corresponding to the 1/32, 1/16, 1/8, and 1/4 guard interval ratio, respectively. In this example, the guard interval ratio is 1/16, so the guard interval is 128.

2. All parameters are related to the guard interval (except TDMA channel parameters). If the guard interval is changed, parameters will be changed to the corresponding values.

3. The TDMA channel condition varies according to channel parameter modification. Parameters include Type, Pathloss, Env, Delay, and Test in the PropNADCtdma model, and Vx and Vy in the AntMobile model.

Simulation Results

Figure 9-40. ML magnitude which shows magnitude of ML function of received signal used to find FFT start. Maximum value appears in every (2048+128) points, the point of the maximum magnitude value is the FFT start.
Figure 9-41. Received 64-QAM Constellation after OFDM Demodulation by TDMA channel. Results show OFDM System Performance.

Figure 9-42. Received DQPSK Constellation after OFDM Demodulation by TDMA Channel. Results show OFDM system performance.

Figure 9-43. Received 16-QAM Constellation after OFDM Demodulation by TDMA Channel.
Figure 9-44. Spectrum of Signal Received from Wireless Channel.  
Center frequency = 611MHz

\[
\begin{array}{|c|c|}
\hline
& \text{ACPR} \\
\hline
\text{ACPR}(1) & -41.099 \\
\text{ACPR}(2) & -48.302 \\
\hline
\end{array}
\]

Figure 9-45. Adjacent-Channel Power Ratio.  
equation is:

\[\text{ACPR} = \text{acpr}_\text{vr}(\text{DsnDTV}_\text{ISDBOFDM}_\text{ThrLay}.\text{RxSignal}, 50.0,\{2.808\text{MHz},2.808\text{MHz}\}, \{8.808\text{MHz},-3.192\text{MHz}\}, \{3.192\text{MHz},8.808\text{MHz}\})\]

Benchmark

- Hardware Platform: Pentium II 200 MHz, 96 MB memory
- Software Platform: WindowsNT 4.0 Workstation, ADS 1.3
- Data Points: 10 symbols
- Simulation Time: 500 seconds

References

[1] ARIB-J APAN, “Terrestrial Integrated Services Digital Broadcasting (ISDB-T);  
Specification of Channel Coding, Framing Structure and Modulation,”  
DTV ISDB System

OFDM 64-QAM ISDB-T System without Channel Coding BER

DTV_ISDBSystem_prj Design Name
DsnDTV_ISDBOFDM_64QAM_BER.dsn

Features

• 64-QAM modulation and demodulation
• Gaussian simulation channels
• Bit error rate is tested
• Without channel coding and interleaving
• Received signal constellation is displayed

Description

This design example is an OFDM adaptation for ISDB-T to test the BER of an ISDB-T system without channel coding. In mode 1, the guard interval ratio is 1/8; modulation mode is 64-QAM.

Simulation results will be compared to those of DsnDTV_ISDBOneLay_64QAM.dsn, which tests the BER with channel coding and interleaving.

Schematics

Figure 9-46 shows the schematic for this design; subnetwork designs are shown in Figure 9-47 and Figure 9-48.
ISDB-T Design Examples

Figure 9-46. DsnDTV_ISDBOFDM_64QAM_BER.dsn

Figure 9-47. sub_ISDBOFDM_CohMod.dsn

Figure 9-48. sub_ISDBOFDM_CohDemod.dsn

9-32 DTV ISDB System
Specifications

<table>
<thead>
<tr>
<th>Specification (Parameter)</th>
<th>Simulation Type</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Carriers</td>
<td>ADS Ptolemy</td>
<td>108</td>
</tr>
<tr>
<td>IFFTSize</td>
<td>ADS Ptolemy</td>
<td>2048</td>
</tr>
<tr>
<td>FFTSize</td>
<td>ADS Ptolemy</td>
<td>2048</td>
</tr>
<tr>
<td>Guard</td>
<td>ADS Ptolemy</td>
<td>256</td>
</tr>
<tr>
<td>IFFTOrder</td>
<td>ADS Ptolemy</td>
<td>11</td>
</tr>
<tr>
<td>FFTOrder</td>
<td>ADS Ptolemy</td>
<td>11</td>
</tr>
<tr>
<td>MaxDelay</td>
<td>ADS Ptolemy</td>
<td>2047</td>
</tr>
</tbody>
</table>

Notes

1. The modification of the guard interval is according to the ISDB-T. In mode 1, the value of the guard interval is 64, 128, 256, and 512, corresponding to the 1/32, 1/16, 1/8, and 1/4 guard interval ratio, respectively. In this example, the guard interval ratio is 1/8, so the guard interval is 256.

2. This design uses the AWGN channel. The DsnDTV_ISDBOFDM_64QAM_BER_AWGN.ds is the simulation of Gaussian channel (AWGN).

Simulation Results

Figure 9-49 shows the received constellation after OFDM demodulation when CN is 15dB and 25dB in Gaussian channel simulation. The results show the higher CN is better than the lower.

Figure 9-50 shows Gaussian channel BER of different CN.
ISDB-T Design Examples

Figure 9-49. Constellation of OFDM Demodulation Signal at Different CN

Figure 9-50. Gaussian Channel BER
Benchmark

- Hardware Platform: Pentium II 200 MHz, 128 MB memory
- Software Platform: WindowsNT 4.0 Workstation, ADS 1.3
- Data Points: 200 OFDM symbols
- Simulation Time: 58 minutes

References

OFDM DQPSK ISDB-T System without Channel Coding BER

DTV_ISDBSystem_prj Design Name
DsnDTV_ISDBOFDM_DQPSK_BER.dsn

Features

• DQPSK modulation and demodulation
• Gaussian simulation channels
• Bit error rate is tested
• Without channel coding and interleaving
• Received signal constellation is displayed

Description

This design example is an OFDM adaptation for ISDB-T to test the BER of an ISDB-T system without channel coding. In mode 1, the guard interval ratio is 1/8; modulation mode is DQPSK.

Simulation results are compared to those of DsnDTV_ISDBOneLay_DQPSK.dsn, which tests the BER with channel coding and interleaving.

Schematics

Figure 9-51 shows the schematic for this design; subnetwork designs are shown in Figure 9-52 and Figure 9-53.
Figure 9-51. DsnDTV_ISDBOFDM_DQPSK_BER.dsn

Figure 9-52. sub_ISDBOFDM_CohMod.dsn

Figure 9-53. sub_ISDBOFDM_CohDemod.dsn
ISDB-T Design Examples

Specifications

<table>
<thead>
<tr>
<th>Specification (Parameter)</th>
<th>Simulation Type</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Carriers</td>
<td>ADS Ptolemy</td>
<td>108</td>
</tr>
<tr>
<td>IFFTSize</td>
<td>ADS Ptolemy</td>
<td>2048</td>
</tr>
<tr>
<td>FFTSize</td>
<td>ADS Ptolemy</td>
<td>2048</td>
</tr>
<tr>
<td>Guard</td>
<td>ADS Ptolemy</td>
<td>256</td>
</tr>
<tr>
<td>IFFTOrder</td>
<td>ADS Ptolemy</td>
<td>11</td>
</tr>
<tr>
<td>FFTOrder</td>
<td>ADS Ptolemy</td>
<td>11</td>
</tr>
<tr>
<td>MaxDelay</td>
<td>ADS Ptolemy</td>
<td>2047</td>
</tr>
</tbody>
</table>

Notes

1. The modification of the guard interval is according to the ISDB-T. In mode 1, the value of the guard interval is 64, 128, 256, and 512 corresponding to the 1/32, 1/16, 1/8, and 1/4 guard interval ratio, respectively. In this example, the guard interval ratio is 1/8, so the guard interval is 256.

2. This design will work in AWGN channel. The DsnDTV_ISDBOFDM_DQPSK_BER_AWGN.ds is the simulation of Gaussian channel (AWGN).

Simulation Results

Figure 9-54 shows the received constellation after OFDM demodulation when the CN is 15dB and 25dB, respectively in the Gaussian channel simulation. The results show the result of higher CN is better than that of lower CN.

Figure 9-55 shows the BER of different CN when the simulation channel is Gaussian channel.
Figure 9-54. Constellation of the OFDM Demodulation Signal at Different CN

Figure 9-55. Gaussian Channel BER
ISDB-T Design Examples

Benchmark

- Hardware Platform: Pentium II 200 MHz, 128 MB memory
- Software Platform: WindowsNT 4.0 Workstation, ADS 1.3
- Data Points: 400 OFDM symbols
- Simulation Time: 2 hours

References

1-Layer 64-QAM Mapping ISDB-T System Design

DTV_ISDBSystem_prj Design Name
DsnDTV_ISDBOneLay_64QAM.dsn

Features

- 1-layer full ISDB system with channel coding/decoding and OFDM modulation/demodulation
- OFDM modulation mode 1, 96 carriers per OFDM segment
- 64-QAM mapping
- 3/4 rate punctured convolutional coding and Viterbi decoding
- Reed-Solomon coding and decoding
- Bytewise interleaving, time and frequency interleaving
- Carrier rotation and scrambling
- AWGN and multipath channel simulation

Description

This example demonstrates the functionality of a full 1-layer ISDB system, including channel coding/decoding and OFDM modulation/demodulation models.

This design is simulated under AWGN and multipath channels. The bit error rates of these channels are shown in the simulation results. In the AWGN channel, the BER is compared to that of DsnDTV_ISDBOFDM_64QAM_BER.dsn without channel coding and interleaving.

Schematics

Figure 9-56 shows the schematic for this design; the coding and decoding subnetwork designs are shown in Figure 9-57 through Figure 9-60.
ISDB-T Design Examples

Figure 9-56. DsnDTV_ISDBOneLay_64QAM.dsn

Figure 9-57. sub_ISDBChCoder_64QAM_3_4.dsn

Figure 9-58. sub_ISDBChDecoder_64QAM_3_4.dsn
Specifications

<table>
<thead>
<tr>
<th>Specification</th>
<th>Simulation Type</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of carriers in each OFDM segment (mode 1)</td>
<td>ADS Ptolemy</td>
<td>96</td>
</tr>
<tr>
<td>Number of OFDM segments in this layer</td>
<td>ADS Ptolemy</td>
<td>13</td>
</tr>
<tr>
<td>Punctured convolutional code rate</td>
<td>ADS Ptolemy</td>
<td>3/4</td>
</tr>
<tr>
<td>Constellation mapping</td>
<td>ADS Ptolemy</td>
<td>64QAM</td>
</tr>
</tbody>
</table>

Notes

1. In this design, some system parameters are represented by global variables in order to adapt the system to different modulation modes. These variables are listed in Table 9-1.

2. Delay is introduced in the system in several places as listed in Table 9-2.

The following parameter settings are important because they relate to the delays in Table 9-2.

- In the DTV_PuncConvDecoder model, OFDM symbol and Viterbi decoding delays are summed and adjusted to a multiple of 204 bytes in order to correctly align the data packet for Reed-Solomon decoding.

- In the sub_ISDBDerandomize subnetwork, delay parameters must be set correctly for the pseudo random sequence to be reset at the right timing.
ISDB-T Design Examples

- The sink of the ultimate output of the system must take into account all delays for correct output data that corresponds to system input.

### Table 9-1. Global Variables

<table>
<thead>
<tr>
<th>Variables</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Carriers</td>
<td>Number of carriers per segment in different OFDM modes: 96, 192, or 384 for modes 1, 2, or 3, respectively.</td>
</tr>
<tr>
<td>Segments</td>
<td>Number of OFDM segments belonging to this layer; in this 1-layer application, it is always 13.</td>
</tr>
<tr>
<td>NumberTSP</td>
<td>Number of TSPs transmitted on all carriers of each OFDM segment. It is also determined by the data rate of punctured convolutional code type and constellation mapping.</td>
</tr>
<tr>
<td>FrameDelays</td>
<td>Number of TSPs delayed in the entire system caused by byte interleaving (1 OFDM frame delay) and time interleaving (several OFDM frame delays, dependent on the I parameter of time interleaving. In this example, I=4; therefore, FrameDelays is 2 OFDM frames).</td>
</tr>
<tr>
<td>SymDelays</td>
<td>Number of bytes delayed in the entire system caused by bit interleaving (2 OFDM symbols) and OFDM demodulation (1 OFDM symbol).</td>
</tr>
<tr>
<td>CN</td>
<td>Carrier/noise ratio of the channel.</td>
</tr>
</tbody>
</table>

### Table 9-2. Delay Adjustments

<table>
<thead>
<tr>
<th>Delay Type</th>
<th>Delay Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>Byte interleaving and delay adjustment</td>
<td>1 OFDM frame</td>
</tr>
<tr>
<td>Bit interleaving and delay adjustment</td>
<td>2 OFDM symbols</td>
</tr>
<tr>
<td>Time interleaving and delay adjustment</td>
<td>Several OFDM frames (depending on the time interleaving mode)</td>
</tr>
<tr>
<td>OFDM demodulation</td>
<td>1 OFDM symbol</td>
</tr>
<tr>
<td>Viterbi decoding</td>
<td>several bytes determined by PathLen</td>
</tr>
</tbody>
</table>

Simulation Results

- AWGN channel

Additive white Gaussian noise was added to set the carrier-to-noise ratio (CN) at the input of the receiver. Figure 9-61 show the transmitted and the received byte data when CN is 15dB and 20dB in the Gaussian channel simulation. The second column is the received byte of CN=15dB; the third column is the received Byte of CN=20dB. Compared to TxByte, the second column has more errors than the third column.

The BERs of the full system before RS decoding and OFDM system without channel coding (in DsnDTV_ISDBOFDM_64QAM_BER.dsn) are shown in Figure 9-62. Channel coding gain is approximately 6.0dB at 0.0001 BER.
<table>
<thead>
<tr>
<th>Index</th>
<th>...[12]</th>
<th>Index</th>
<th>...[12]</th>
<th>...[12]</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>71.000</td>
<td>71.000</td>
<td>71.000</td>
<td>71.000</td>
</tr>
<tr>
<td>1</td>
<td>192.000</td>
<td>397996</td>
<td>29.000</td>
<td>192.000</td>
</tr>
<tr>
<td>2</td>
<td>194.000</td>
<td>397996</td>
<td>48.000</td>
<td>194.000</td>
</tr>
<tr>
<td>3</td>
<td>3.000</td>
<td>397995</td>
<td>3.000</td>
<td>3.000</td>
</tr>
<tr>
<td>4</td>
<td>160.000</td>
<td>398000</td>
<td>116.000</td>
<td>180.000</td>
</tr>
<tr>
<td>5</td>
<td>192.000</td>
<td>398001</td>
<td>8.000</td>
<td>192.000</td>
</tr>
<tr>
<td>6</td>
<td>58.000</td>
<td>398002</td>
<td>77.000</td>
<td>55.000</td>
</tr>
<tr>
<td>7</td>
<td>160.000</td>
<td>398003</td>
<td>160.000</td>
<td>180.000</td>
</tr>
<tr>
<td>8</td>
<td>3.000</td>
<td>398004</td>
<td>8.000</td>
<td>8.000</td>
</tr>
<tr>
<td>9</td>
<td>78.000</td>
<td>398005</td>
<td>28.000</td>
<td>28.000</td>
</tr>
<tr>
<td>10</td>
<td>52.000</td>
<td>398006</td>
<td>52.000</td>
<td>52.000</td>
</tr>
<tr>
<td>11</td>
<td>14.000</td>
<td>398007</td>
<td>14.000</td>
<td>14.000</td>
</tr>
<tr>
<td>12</td>
<td>251.000</td>
<td>398008</td>
<td>251.000</td>
<td>251.000</td>
</tr>
<tr>
<td>13</td>
<td>204.000</td>
<td>398009</td>
<td>89.000</td>
<td>204.000</td>
</tr>
<tr>
<td>14</td>
<td>4.000</td>
<td>398010</td>
<td>6.000</td>
<td>4.000</td>
</tr>
<tr>
<td>15</td>
<td>113.000</td>
<td>398011</td>
<td>98.000</td>
<td>113.000</td>
</tr>
<tr>
<td>16</td>
<td>228.000</td>
<td>398012</td>
<td>228.000</td>
<td>228.000</td>
</tr>
<tr>
<td>17</td>
<td>203.000</td>
<td>398013</td>
<td>162.000</td>
<td>202.000</td>
</tr>
<tr>
<td>18</td>
<td>122.000</td>
<td>398014</td>
<td>122.000</td>
<td>122.000</td>
</tr>
<tr>
<td>19</td>
<td>170.000</td>
<td>398015</td>
<td>170.000</td>
<td>170.000</td>
</tr>
<tr>
<td>20</td>
<td>38.000</td>
<td>398016</td>
<td>38.000</td>
<td>38.000</td>
</tr>
<tr>
<td>21</td>
<td>182.000</td>
<td>398017</td>
<td>182.000</td>
<td>182.000</td>
</tr>
<tr>
<td>22</td>
<td>232.000</td>
<td>398016</td>
<td>232.000</td>
<td>232.000</td>
</tr>
<tr>
<td>23</td>
<td>32.000</td>
<td>398019</td>
<td>32.000</td>
<td>32.000</td>
</tr>
<tr>
<td>24</td>
<td>87.000</td>
<td>398020</td>
<td>87.000</td>
<td>87.000</td>
</tr>
<tr>
<td>25</td>
<td>151.000</td>
<td>398021</td>
<td>23.000</td>
<td>191.000</td>
</tr>
<tr>
<td>26</td>
<td>220.000</td>
<td>398022</td>
<td>220.000</td>
<td>220.000</td>
</tr>
<tr>
<td>27</td>
<td>7.000</td>
<td>398023</td>
<td>7.000</td>
<td>7.000</td>
</tr>
<tr>
<td>28</td>
<td>88.000</td>
<td>398024</td>
<td>88.000</td>
<td>88.000</td>
</tr>
<tr>
<td>29</td>
<td>87.000</td>
<td>398025</td>
<td>87.000</td>
<td>87.000</td>
</tr>
<tr>
<td>30</td>
<td>103.000</td>
<td>398026</td>
<td>103.000</td>
<td>103.000</td>
</tr>
<tr>
<td>31</td>
<td>251.000</td>
<td>398027</td>
<td>251.000</td>
<td>251.000</td>
</tr>
<tr>
<td>32</td>
<td>183.000</td>
<td>398028</td>
<td>183.000</td>
<td>183.000</td>
</tr>
<tr>
<td>33</td>
<td>91.000</td>
<td>398029</td>
<td>91.000</td>
<td>91.000</td>
</tr>
<tr>
<td>34</td>
<td>175.000</td>
<td>398030</td>
<td>175.000</td>
<td>175.000</td>
</tr>
<tr>
<td>35</td>
<td>38.000</td>
<td>398031</td>
<td>89.000</td>
<td>38.000</td>
</tr>
<tr>
<td>36</td>
<td>216.000</td>
<td>398032</td>
<td>216.000</td>
<td>216.000</td>
</tr>
<tr>
<td>37</td>
<td>187.000</td>
<td>398033</td>
<td>116.000</td>
<td>187.000</td>
</tr>
<tr>
<td>38</td>
<td>19.000</td>
<td>398034</td>
<td>19.000</td>
<td>19.000</td>
</tr>
<tr>
<td>39</td>
<td>27.000</td>
<td>398035</td>
<td>6.000</td>
<td>27.000</td>
</tr>
</tbody>
</table>

Figure 9-61. Compare RxByte with TxByte with different CN
Measurements of BER vs. CN were made using a fading channel simulator. The DU ratios of a main signal and a delayed signal were set to 3 dB. Delay time of a delayed signal from a main signal was set to 1.5 $\mu$sec. This multipath channel has no fading frequency. In the ADS simulation system, this multipath channel is simulated by UserDefChannel model; parameters were set as follows:

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Specification</th>
<th>Simulation Type</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>UserDefChannel</td>
<td>PathNumber</td>
<td>ADS Ptolemy</td>
<td>2</td>
</tr>
<tr>
<td>UserDefChannel</td>
<td>AmpArray</td>
<td>ADS Ptolemy</td>
<td>“0.8165 0.5773”</td>
</tr>
<tr>
<td>UserDefChannel</td>
<td>DelayArray</td>
<td>ADS Ptolemy</td>
<td>“0.0 1.5”</td>
</tr>
<tr>
<td>AntMobile</td>
<td>Vx</td>
<td>ADS Ptolemy</td>
<td>0</td>
</tr>
</tbody>
</table>

This simulation result was not included into this DTV package because of the limitation of DTV package size. To prove the BER performance of this channel, set the parameters above and run this design example. The simulation time is very long.

Figure 9-63 shows the transmitted and received Byte data when carrier-to-noise ratio (CN) is 15dB and 25dB in the multipath channel.
simulation. In Figure 9-63, the second column is the received Byte of the CN=15dB, the third column is the received Byte of 25dB. Compared to TxByte, the second column has more error than the third column.

The BER of the full system before RS decoding is shown in Figure 9-64. CN is 25.0dB when BER is 0.0001.

<table>
<thead>
<tr>
<th>Index</th>
<th>...</th>
<th>...</th>
<th>...</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>72.0</td>
<td>38798</td>
<td>81.0</td>
<td>71.0</td>
</tr>
<tr>
<td>1</td>
<td>155.0</td>
<td>38797</td>
<td>155.0</td>
<td>195.0</td>
</tr>
<tr>
<td>2</td>
<td>152.0</td>
<td>38798</td>
<td>152.0</td>
<td>162.0</td>
</tr>
<tr>
<td>3</td>
<td>159.0</td>
<td>38795</td>
<td>25.0</td>
<td>189.0</td>
</tr>
<tr>
<td>4</td>
<td>46.0</td>
<td>38800</td>
<td>102.0</td>
<td>49.0</td>
</tr>
<tr>
<td>5</td>
<td>27.0</td>
<td>38801</td>
<td>27.0</td>
<td>27.0</td>
</tr>
<tr>
<td>6</td>
<td>211.0</td>
<td>38802</td>
<td>4.0</td>
<td>211.0</td>
</tr>
<tr>
<td>7</td>
<td>333.0</td>
<td>38803</td>
<td>233.0</td>
<td>333.0</td>
</tr>
<tr>
<td>8</td>
<td>80.0</td>
<td>38804</td>
<td>80.0</td>
<td>80.0</td>
</tr>
<tr>
<td>9</td>
<td>123.0</td>
<td>38805</td>
<td>123.0</td>
<td>123.0</td>
</tr>
<tr>
<td>10</td>
<td>193.0</td>
<td>38806</td>
<td>216.0</td>
<td>193.0</td>
</tr>
<tr>
<td>11</td>
<td>187.0</td>
<td>38807</td>
<td>187.0</td>
<td>187.0</td>
</tr>
<tr>
<td>12</td>
<td>171.0</td>
<td>38808</td>
<td>171.0</td>
<td>171.0</td>
</tr>
<tr>
<td>13</td>
<td>245.0</td>
<td>38809</td>
<td>128.0</td>
<td>245.0</td>
</tr>
<tr>
<td>14</td>
<td>31.0</td>
<td>38810</td>
<td>37.0</td>
<td>31.0</td>
</tr>
<tr>
<td>15</td>
<td>188.0</td>
<td>38811</td>
<td>188.0</td>
<td>188.0</td>
</tr>
<tr>
<td>16</td>
<td>100.0</td>
<td>38812</td>
<td>9.0</td>
<td>100.0</td>
</tr>
<tr>
<td>17</td>
<td>104.0</td>
<td>38813</td>
<td>81.0</td>
<td>104.0</td>
</tr>
<tr>
<td>18</td>
<td>236.0</td>
<td>38814</td>
<td>187.0</td>
<td>236.0</td>
</tr>
<tr>
<td>19</td>
<td>111.0</td>
<td>38815</td>
<td>181.0</td>
<td>111.0</td>
</tr>
<tr>
<td>20</td>
<td>36.0</td>
<td>38816</td>
<td>36.0</td>
<td>36.0</td>
</tr>
<tr>
<td>21</td>
<td>157.0</td>
<td>38817</td>
<td>157.0</td>
<td>157.0</td>
</tr>
<tr>
<td>22</td>
<td>59.0</td>
<td>38818</td>
<td>59.0</td>
<td>59.0</td>
</tr>
<tr>
<td>23</td>
<td>59.0</td>
<td>38819</td>
<td>59.0</td>
<td>59.0</td>
</tr>
<tr>
<td>24</td>
<td>18.0</td>
<td>38820</td>
<td>18.0</td>
<td>18.0</td>
</tr>
<tr>
<td>25</td>
<td>196.0</td>
<td>38821</td>
<td>11.0</td>
<td>196.0</td>
</tr>
<tr>
<td>26</td>
<td>31.0</td>
<td>38822</td>
<td>19.0</td>
<td>31.0</td>
</tr>
<tr>
<td>27</td>
<td>217.0</td>
<td>38823</td>
<td>217.0</td>
<td>217.0</td>
</tr>
<tr>
<td>28</td>
<td>122.0</td>
<td>38824</td>
<td>4.0</td>
<td>122.0</td>
</tr>
<tr>
<td>29</td>
<td>86.0</td>
<td>38825</td>
<td>86.0</td>
<td>86.0</td>
</tr>
<tr>
<td>30</td>
<td>12.0</td>
<td>38826</td>
<td>12.0</td>
<td>12.0</td>
</tr>
<tr>
<td>31</td>
<td>209.0</td>
<td>38827</td>
<td>209.0</td>
<td>209.0</td>
</tr>
<tr>
<td>32</td>
<td>161.0</td>
<td>38828</td>
<td>249.0</td>
<td>161.0</td>
</tr>
<tr>
<td>33</td>
<td>14.0</td>
<td>38829</td>
<td>14.0</td>
<td>14.0</td>
</tr>
<tr>
<td>34</td>
<td>10.0</td>
<td>38830</td>
<td>10.0</td>
<td>10.0</td>
</tr>
<tr>
<td>35</td>
<td>25.0</td>
<td>38831</td>
<td>25.0</td>
<td>25.0</td>
</tr>
<tr>
<td>36</td>
<td>215.0</td>
<td>38832</td>
<td>215.0</td>
<td>215.0</td>
</tr>
<tr>
<td>37</td>
<td>12.0</td>
<td>38833</td>
<td>12.0</td>
<td>12.0</td>
</tr>
<tr>
<td>38</td>
<td>322.0</td>
<td>38834</td>
<td>222.0</td>
<td>222.0</td>
</tr>
</tbody>
</table>

Figure 9-63. Compare RxByte with TxByte with different CN
Measurements of BER vs. CN were made using a fading channel simulator. The DU ratios of a main signal and a delayed signal were set to 10 dB. Delay time of a delayed signal from a main signal was set to 1.5 us. This multipath channel has no fading frequency. In the ADS simulation system, this multipath channel is simulated by UserDefChannel model, its parameters were set as follows:

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Specification</th>
<th>Simulation Type</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>UserDefChannel</td>
<td>PathNumber</td>
<td>ADS Ptolemy</td>
<td>2</td>
</tr>
<tr>
<td>UserDefChannel</td>
<td>AmpArray</td>
<td>ADS Ptolemy</td>
<td>0.9535 0.3015</td>
</tr>
<tr>
<td>UserDefChannel</td>
<td>DelayArray</td>
<td>ADS Ptolemy</td>
<td>0.0 1.5</td>
</tr>
<tr>
<td>AntMobile</td>
<td>Vx</td>
<td>ADS Ptolemy</td>
<td>0</td>
</tr>
</tbody>
</table>

This simulation result was not included into this DTV package because of the limitation of DTV package size. If users want to prove the BER performance of this channel, users can set the parameters above and run this design example. The simulation time is very long.

Figure 9-65 show the transmitted Byte data and the received Byte data when carrier-to-noise ratio (CN) is 15dB and 20dB in the multipath channel simulation. In Figure 9-65, the second column is the received Byte of the CN=15dB, the third column is the received Byte of 20dB. Compared to TxByte, the second column has more errors than the the third column.
The BER of the full system before RS decoding is shown in Figure 9-66. The CN is 20.0dB when BER is 0.0001.

<table>
<thead>
<tr>
<th>Index</th>
<th>...</th>
<th>Index</th>
<th>...</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>21.000</td>
<td>25</td>
<td>79868</td>
<td>71.000</td>
</tr>
<tr>
<td>1</td>
<td>195.000</td>
<td>26</td>
<td>79867</td>
<td>184.000</td>
</tr>
<tr>
<td>2</td>
<td>162.000</td>
<td>27</td>
<td>79868</td>
<td>56.000</td>
</tr>
<tr>
<td>3</td>
<td>189.000</td>
<td>28</td>
<td>79869</td>
<td>233.000</td>
</tr>
<tr>
<td>4</td>
<td>48.000</td>
<td>29</td>
<td>79800</td>
<td>104.000</td>
</tr>
<tr>
<td>5</td>
<td>27.000</td>
<td>30</td>
<td>79801</td>
<td>27.000</td>
</tr>
<tr>
<td>6</td>
<td>211.000</td>
<td>31</td>
<td>79802</td>
<td>147.000</td>
</tr>
<tr>
<td>7</td>
<td>233.000</td>
<td>32</td>
<td>79803</td>
<td>233.000</td>
</tr>
<tr>
<td>8</td>
<td>80.000</td>
<td>33</td>
<td>79804</td>
<td>80.000</td>
</tr>
<tr>
<td>9</td>
<td>122.000</td>
<td>34</td>
<td>79805</td>
<td>111.000</td>
</tr>
<tr>
<td>10</td>
<td>153.000</td>
<td>35</td>
<td>79806</td>
<td>131.000</td>
</tr>
<tr>
<td>11</td>
<td>187.000</td>
<td>36</td>
<td>79807</td>
<td>187.000</td>
</tr>
<tr>
<td>12</td>
<td>177.000</td>
<td>37</td>
<td>79808</td>
<td>115.000</td>
</tr>
<tr>
<td>13</td>
<td>245.000</td>
<td>38</td>
<td>79809</td>
<td>245.000</td>
</tr>
<tr>
<td>14</td>
<td>31.000</td>
<td>39</td>
<td>79810</td>
<td>115.000</td>
</tr>
<tr>
<td>15</td>
<td>153.000</td>
<td>40</td>
<td>79811</td>
<td>189.000</td>
</tr>
<tr>
<td>16</td>
<td>150.000</td>
<td>41</td>
<td>79812</td>
<td>190.000</td>
</tr>
<tr>
<td>17</td>
<td>154.000</td>
<td>42</td>
<td>79813</td>
<td>206.000</td>
</tr>
<tr>
<td>18</td>
<td>236.000</td>
<td>43</td>
<td>79814</td>
<td>49.000</td>
</tr>
<tr>
<td>19</td>
<td>111.000</td>
<td>44</td>
<td>79815</td>
<td>171.000</td>
</tr>
<tr>
<td>20</td>
<td>39.000</td>
<td>45</td>
<td>79819</td>
<td>36.000</td>
</tr>
<tr>
<td>21</td>
<td>187.000</td>
<td>46</td>
<td>79817</td>
<td>167.000</td>
</tr>
<tr>
<td>22</td>
<td>55.000</td>
<td>47</td>
<td>79818</td>
<td>55.000</td>
</tr>
<tr>
<td>23</td>
<td>68.000</td>
<td>48</td>
<td>79819</td>
<td>68.000</td>
</tr>
<tr>
<td>24</td>
<td>188.000</td>
<td>49</td>
<td>79820</td>
<td>180.000</td>
</tr>
<tr>
<td>25</td>
<td>158.000</td>
<td>50</td>
<td>79821</td>
<td>198.000</td>
</tr>
<tr>
<td>26</td>
<td>9.000</td>
<td>51</td>
<td>79822</td>
<td>19.000</td>
</tr>
<tr>
<td>27</td>
<td>217.000</td>
<td>52</td>
<td>79823</td>
<td>217.000</td>
</tr>
<tr>
<td>28</td>
<td>152.000</td>
<td>53</td>
<td>79824</td>
<td>152.000</td>
</tr>
<tr>
<td>29</td>
<td>68.000</td>
<td>54</td>
<td>79825</td>
<td>68.000</td>
</tr>
<tr>
<td>30</td>
<td>12.000</td>
<td>55</td>
<td>79826</td>
<td>27.000</td>
</tr>
<tr>
<td>31</td>
<td>209.000</td>
<td>56</td>
<td>79827</td>
<td>143.000</td>
</tr>
<tr>
<td>32</td>
<td>101.000</td>
<td>57</td>
<td>79828</td>
<td>190.000</td>
</tr>
<tr>
<td>33</td>
<td>14.000</td>
<td>58</td>
<td>79829</td>
<td>48.000</td>
</tr>
<tr>
<td>34</td>
<td>13.000</td>
<td>59</td>
<td>79830</td>
<td>13.000</td>
</tr>
<tr>
<td>35</td>
<td>29.000</td>
<td>60</td>
<td>79831</td>
<td>77.000</td>
</tr>
<tr>
<td>36</td>
<td>215.000</td>
<td>61</td>
<td>79832</td>
<td>215.000</td>
</tr>
<tr>
<td>37</td>
<td>52.000</td>
<td>62</td>
<td>79833</td>
<td>52.000</td>
</tr>
<tr>
<td>38</td>
<td>247.000</td>
<td>63</td>
<td>79834</td>
<td>54.000</td>
</tr>
<tr>
<td>39</td>
<td>16.000</td>
<td>64</td>
<td>79835</td>
<td>115.000</td>
</tr>
</tbody>
</table>

Figure 9-65. Compare RxByte with TxByte with different CN
Figure 9-67 shows the BERs of the three different simulation results. The performance of AWGN channel is better than the multipath channels (DU ratios are 3dB and 10dB); the performance of the 10dB multipath channel is better than the 3dB multipath channel.
Benchmark

- Hardware Platform: Pentium II 200 MHz, 128 MB memory
- Software Platform: WindowsNT 4.0 Workstation, ADS 1.3
- Data Points: 188*5-1
- Simulation Time: 16 hours for AWGN channel, 20 hours for DU = 3 dB and DU = 10 dB multipath channel.

References

ISDB-T Design Examples

1-Layer DQPSK-Mapping ISDB-T System Design

DTV_ISDBSystem_prj Design Name
DsnDTV_ISDBOneLay_DQPSK.dsn

Features

• 1-layer full ISDB system with channel coding/decoding and OFDM modulation/demodulation
• OFDM modulation mode 1, 96 carriers per OFDM segment
• DQPSK mapping
• 1/2 rate punctured convolutional coding
• Reed-Solomon coding and decoding
• Bytewise interleaving, time and frequency interleaving
• Carrier rotation and scrambling
• AWGN, multipath, and Rayleigh fading channel simulations

Description

This design demonstrates the performance of a full 1-layer ISDB system, including channel coding/decoding and OFDM modulation/demodulation models. It uses OFDM mode 1 and OFDM modulation, DQPSK mapping, 1/2 rate punctured convolutional coding, interleaving and delay adjustments.

This design is simulated under AWGN, multipath, and Rayleigh fading channels. The bit error rates of these channels are shown in the simulation results. In the AWGN channel, the BER is compared to the DsnDTV_ISDBOFDM_DQPSK_BER.dsn design without channel coding and interleaving.

Schematics

Figure 9-68 shows the schematic for this design; the coding and decoding subnetwork designs are shown in Figure 9-69 through Figure 9-72.
Figure 9-68. DsnDTV_ISDBOneLay_DQPSK.dsn

Figure 9-69. sub_ISDBChCoder_DQPSK_1_2.dsn

Figure 9-70. sub_ISDBChDecoder_DQPSK_1_2.dsn
1. In this design, some system parameters are represented by global variables in order to adapt the system to different modulation modes. These variables are listed in Table 9-3.

2. Delay is introduced in the system in several places as listed in Table 9-4.

   The following parameter settings are important as they relate to the delays in Table 9-4.

   • In the DTV_PuncConvDecoder model, OFDM symbol and Viterbi decoding delays are summed and adjusted to a multiple of 204 bytes in order to correctly align the data packet for Reed-Solomon decoding.

   ![Diagram](image1)

   **Figure 9-71. sub_ISDBOFDM_CohMod.dsn**

   ![Diagram](image2)

   **Figure 9-72. sub_ISDBOFDM_CohDemod.dsn**
• In the sub_ISDBDerandomize subnetwork, delay parameters must be set correctly for the pseudo random sequence to be reset at the right timing.
• The sink of the ultimate output of the system must take into account all delays for correct output data that corresponds to system input.

Table 9-3. Global Variables

<table>
<thead>
<tr>
<th>Variables</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Carriers</td>
<td>Number of carriers per segment in different OFDM modes: 96, 192, or 384 for modes 1, 2, or 3, respectively.</td>
</tr>
<tr>
<td>Segments</td>
<td>Number of OFDM segments belonging to this layer; in this 1-layer application, it is always 13.</td>
</tr>
<tr>
<td>NumberTSP</td>
<td>Number of TSPs transmitted on all carriers of each OFDM segment. It is also determined by the data rate of punctured convolutional code type and constellation mapping.</td>
</tr>
<tr>
<td>FrameDelays</td>
<td>Number of TSPs delayed in the entire system caused by byte interleaving (1 OFDM frame delay) and time interleaving (several OFDM frame delays, dependent on the I parameter of time interleaving. In this example, I=4; therefore, FrameDelays is 2 OFDM frames).</td>
</tr>
<tr>
<td>SymDelays</td>
<td>Number of bytes delayed in the entire system caused by bit interleaving (2 OFDM symbols) and OFDM demodulation (1 OFDM symbol).</td>
</tr>
<tr>
<td>CN</td>
<td>Carrier/noise ratio of the channel.</td>
</tr>
</tbody>
</table>

Table 9-4.

<table>
<thead>
<tr>
<th>Delay Type</th>
<th>Delay Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>Byte interleaving and delay adjustment</td>
<td>1 OFDM frame</td>
</tr>
<tr>
<td>Bit interleaving and delay adjustment</td>
<td>2 OFDM symbols</td>
</tr>
<tr>
<td>Time interleaving and delay adjustment</td>
<td>Several OFDM frames (depending on the time interleaving mode)</td>
</tr>
<tr>
<td>OFDM demodulation</td>
<td>1 OFDM symbol</td>
</tr>
<tr>
<td>Viterbi decoding</td>
<td>several bytes determined by PathLen</td>
</tr>
</tbody>
</table>
Simulated Results

AWGN Channel

Additive white Gaussian noise was added to set the carrier-to-noise ratio (CN) at the input of the receiver. Figure 9-73 shows the transmitted and the received byte data when carrier-to-noise ratio (CN) is 5dB and 7dB in the Gaussian channel simulation.

The BERs of the full system before RS decoding and OFDM system without channel coding (in DsnDTV_ISDBOFDM_DQPSK_BER.dsn) are shown in Figure 9-74. From Figure 9-74, we can know the channel coding gain is about 8.0dB at 0.0001 BER.

<table>
<thead>
<tr>
<th>Index</th>
<th>...</th>
<th>Index</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>250</td>
<td>111.000</td>
<td>288</td>
<td>111.000</td>
</tr>
<tr>
<td>251</td>
<td>11.000</td>
<td>289</td>
<td>111.000</td>
</tr>
<tr>
<td>252</td>
<td>58.000</td>
<td>290</td>
<td>58.000</td>
</tr>
<tr>
<td>253</td>
<td>254.000</td>
<td>291</td>
<td>254.000</td>
</tr>
<tr>
<td>254</td>
<td>250.000</td>
<td>292</td>
<td>250.000</td>
</tr>
<tr>
<td>255</td>
<td>104.000</td>
<td>293</td>
<td>104.000</td>
</tr>
<tr>
<td>256</td>
<td>57.000</td>
<td>294</td>
<td>57.000</td>
</tr>
<tr>
<td>257</td>
<td>122.000</td>
<td>295</td>
<td>122.000</td>
</tr>
<tr>
<td>258</td>
<td>192.000</td>
<td>296</td>
<td>192.000</td>
</tr>
<tr>
<td>259</td>
<td>172.000</td>
<td>297</td>
<td>172.000</td>
</tr>
<tr>
<td>260</td>
<td>158.000</td>
<td>298</td>
<td>158.000</td>
</tr>
<tr>
<td>261</td>
<td>159.000</td>
<td>299</td>
<td>159.000</td>
</tr>
<tr>
<td>262</td>
<td>224.000</td>
<td>300</td>
<td>224.000</td>
</tr>
<tr>
<td>263</td>
<td>98.000</td>
<td>301</td>
<td>98.000</td>
</tr>
<tr>
<td>264</td>
<td>94.000</td>
<td>302</td>
<td>94.000</td>
</tr>
<tr>
<td>265</td>
<td>0.000</td>
<td>303</td>
<td>0.000</td>
</tr>
<tr>
<td>266</td>
<td>87.000</td>
<td>304</td>
<td>87.000</td>
</tr>
<tr>
<td>267</td>
<td>29.000</td>
<td>305</td>
<td>29.000</td>
</tr>
<tr>
<td>268</td>
<td>229.000</td>
<td>306</td>
<td>229.000</td>
</tr>
<tr>
<td>269</td>
<td>172.000</td>
<td>307</td>
<td>172.000</td>
</tr>
<tr>
<td>300</td>
<td>3.000</td>
<td>308</td>
<td>3.000</td>
</tr>
<tr>
<td>301</td>
<td>238.000</td>
<td>309</td>
<td>238.000</td>
</tr>
<tr>
<td>302</td>
<td>251.000</td>
<td>310</td>
<td>251.000</td>
</tr>
<tr>
<td>303</td>
<td>146.000</td>
<td>311</td>
<td>146.000</td>
</tr>
<tr>
<td>304</td>
<td>25.000</td>
<td>312</td>
<td>25.000</td>
</tr>
<tr>
<td>305</td>
<td>49.000</td>
<td>313</td>
<td>49.000</td>
</tr>
<tr>
<td>306</td>
<td>91.000</td>
<td>314</td>
<td>91.000</td>
</tr>
<tr>
<td>307</td>
<td>22.000</td>
<td>315</td>
<td>22.000</td>
</tr>
<tr>
<td>308</td>
<td>4.000</td>
<td>316</td>
<td>4.000</td>
</tr>
<tr>
<td>309</td>
<td>235.000</td>
<td>317</td>
<td>235.000</td>
</tr>
<tr>
<td>310</td>
<td>180.000</td>
<td>318</td>
<td>180.000</td>
</tr>
<tr>
<td>311</td>
<td>90.000</td>
<td>319</td>
<td>90.000</td>
</tr>
<tr>
<td>312</td>
<td>249.000</td>
<td>320</td>
<td>249.000</td>
</tr>
<tr>
<td>313</td>
<td>133.000</td>
<td>321</td>
<td>133.000</td>
</tr>
<tr>
<td>314</td>
<td>107.000</td>
<td>322</td>
<td>107.000</td>
</tr>
<tr>
<td>315</td>
<td>129.000</td>
<td>323</td>
<td>129.000</td>
</tr>
</tbody>
</table>

Figure 9-73. Compare RxByte with TxByte with different CN
Multipath Channel

Measurements of BER vs. CN were made using a fading channel simulator. The DU ratios of a main signal and a delayed signal were set to 3 dB. Delay time of a delayed signal from a main signal was set to 1.5 µsec. This multipath channel has no fading frequency. In the ADS simulation system, this multipath channel is simulated by UserDefChannel model; parameters were set as follows.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Specification</th>
<th>Simulation Type</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>UserDefChannel</td>
<td>PathNumber</td>
<td>ADS Ptolemy</td>
<td>2</td>
</tr>
<tr>
<td>UserDefChannel</td>
<td>AmpArray</td>
<td>ADS Ptolemy</td>
<td>&quot;0.8165 0.5773&quot;</td>
</tr>
<tr>
<td>UserDefChannel</td>
<td>DelayArray</td>
<td>ADS Ptolemy</td>
<td>&quot;0.0 1.5&quot;</td>
</tr>
<tr>
<td>AntMobile</td>
<td>Vx</td>
<td>ADS Ptolemy</td>
<td>0</td>
</tr>
</tbody>
</table>

Figure 9-75 shows the transmitted Byte data and the received Byte data when carrier-to-noise ratio (CN) is 10dB and 13dB in the multipath channel simulation.
ISDB-T Design Examples

Figure 9-75. Compare RxByte with TxByte with different CN

The BER of the full system before RS decoding is shown in Figure 9-76. The CN is 11.8dB when BER is 0.0002.
Rayleigh Channel

Measurements of BER vs. CN were made using a fading channel simulator. 2-path Rayleigh fading was used and the DU ratio was set to 0 dB. Delay time of a delayed signal from a main signal was set to 1.5 µsec. This multipath channel has a 70-Hz fading frequency. In the ADS simulation system, this multipath channel is simulated by UserDefChannel model; parameters were set as follows:

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Specification</th>
<th>Simulation Type</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>UserDefChannel</td>
<td>PathNumber</td>
<td>ADS Ptolemy</td>
<td>2</td>
</tr>
<tr>
<td>UserDefChannel</td>
<td>AmpArray</td>
<td>ADS Ptolemy</td>
<td>&quot;0.707 0.707&quot;</td>
</tr>
<tr>
<td>UserDefChannel</td>
<td>DelayArray</td>
<td>ADS Ptolemy</td>
<td>&quot;0.0 1.5&quot;</td>
</tr>
<tr>
<td>AntMobile</td>
<td>Vx</td>
<td>ADS Ptolemy</td>
<td>124.0</td>
</tr>
</tbody>
</table>

*Figure 9-77* shows the transmitted and received Byte data when carrier-to-noise ratio (CN) is 5dB and 15dB in the Rayleigh channel simulation. In *Figure 9-77*, the second column is the received Byte of the CN =5dB, the third column is the received Byte of 15dB . Compared to TxByte, the second column has more errors than the third.
The BER of the full system before RS decoding is shown in Figure 9-78. The CN is 14.5 dB when BER is 0.0002.
Figure 9-78. Rayleigh Channel BER

**Benchmark**

- Hardware Platform: Pentium II 200 MHz, 128 MB memory
- Software Platform: WindowsNT 4.0 Workstation, ADS 1.3
- Data Points: 2*188-1 Bytes
- Simulation Time: 16 hours for AWGN channel, 22 hours for multipath channel, 40 hours for Rayleigh channel

**References**

3-Layer ISDB-T System Design

DTV_ISDBSystem_prj Design Name
DsnDTV_ISDBThrLay_System.dsn

Features

- 3-layer system design
- Different modulation mode and code rate for each layer
- Reed-Solomon coding and decoding
- Punctured convolutional coding and decoding
- Mode 1 OFDM adaptation
- Byte-wise interleaving and deinterleaving
- Time interleaving and deinterleaving
- Frequency interleaving and deinterleaving

Description

This design is a 3-layer system design example of ISDB-T. It includes 13 segments (1 segment for Layer A, 5 segments for Layer B, and 7 segments for Layer C). The modulation modes and code rates are 64-QAM, 2/3, and DQPSK, 1/2, and 16-QAM, 3/4 in Layer A, Layer B, and Layer C, respectively.

In the OFDM adaptation, the 3-layer full system design works in mode 1. The IFFT/FFT size is 2048; the guard interval is 1/16 of IFFT size. The simulation channel is the TDMA channel. The byte in transmitter and receiver, the received 64-QAM, DQPSK, and 16-QAM constellations are shown in the simulation results.

Schematics

Figure 9-79 shows the schematic for this design; the subnetwork designs are shown in Figure 9-80 through Figure 9-85.
Figure 9-79. DsnDTV_ISDBThrLay_System.dsn

Figure 9-80. sub_ISDBOFDM_ThrLayMod.dsn
ISDB-T Design Examples

Figure 9-81. sub_ISDBOFDM_ThrLayDemod.dsn

Figure 9-82. sub_ISDBThrLay_ChCoder.dsn
Figure 9-83. sub_ISDBThrLay_ChDecoder.dsn

Figure 9-84. sub_ISDBThrLay_FreqInterlv.dsn
ISDB-T Design Examples

Figure 9-85. sub_ISDBThrLay_FreqDeinterLv.dsn

Specifications

<table>
<thead>
<tr>
<th>Specification (Parameter)</th>
<th>Simulation Type</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Carriers</td>
<td>ADS Ptolemy</td>
<td>96</td>
</tr>
<tr>
<td>Segments</td>
<td>ADS Ptolemy</td>
<td>13</td>
</tr>
<tr>
<td>IFFTSize</td>
<td>ADS Ptolemy</td>
<td>2048</td>
</tr>
<tr>
<td>FFTSize</td>
<td>ADS Ptolemy</td>
<td>2048</td>
</tr>
<tr>
<td>Guard</td>
<td>ADS Ptolemy</td>
<td>128</td>
</tr>
<tr>
<td>IFFTOrder</td>
<td>ADS Ptolemy</td>
<td>11</td>
</tr>
<tr>
<td>FFTOrder</td>
<td>ADS Ptolemy</td>
<td>11</td>
</tr>
<tr>
<td>MaxDelay</td>
<td>ADS Ptolemy</td>
<td>2047</td>
</tr>
<tr>
<td>SegmentsA</td>
<td>ADS Ptolemy</td>
<td>1</td>
</tr>
<tr>
<td>SegmentsB</td>
<td>ADS Ptolemy</td>
<td>5</td>
</tr>
<tr>
<td>SegmentsC</td>
<td>ADS Ptolemy</td>
<td>7</td>
</tr>
<tr>
<td>Ru</td>
<td>ADS Ptolemy</td>
<td>0.9</td>
</tr>
</tbody>
</table>

Notes

1. The propagation channel model used in this example is a standard TDMA channel.
2. The DelayByte parameter of the punctured convolutional decoder model (DTV_PunConvDecoder) is used to adjust delays caused by OFDM demodulation (which is one OFDM symbol) and two additional OFDM symbol delays, which correspond to the bit interleaving procedure because delay adjustment for bit interleaving is 2 OFDM symbols.

3. Total delay caused by byte-wise interleaving and deinterleaving is 17*11*12 bytes (corresponding to 11 TSPs). Because the OFDM symbol is transmitted as an OFDM frame, delay of the TSPs is adjusted to one OFDM frame.

4. Total delay caused by time interleaving and deinterleaving is adjusted in accordance with an integer I as shown in table 4-3 in ISDB-T. Because I = 4 in this 3-layer design, the number of OFDM frames to be delayed by the delay adjustment and time interleaving is 2.

5. The last delay adjusted in this design generates the number of total null TSP bytes which is needed per operation in DTV_SynThreeLayTSP model.

Simulation Results

Figure 9-86 shows the transmitted and received TSP data.
Figure 9-86. Compare RxTSP with TxTSP

Figure 9-87, Figure 9-88, and Figure 9-89 show the received constellations for Layers A, B, and C.
**Figure 9-88. Received Constellation for Layer B**

**Figure 9-89. Received Constellation for Layer C**

**Benchmark**
- Hardware Platform: Pentium II 200 MHz, 128 MB memory
- Software Platform: WindowsNT 4.0 Workstation, ADS 1.3
- Data Points: 188*((12*5+36*7+48*1)*3+40) Bytes
- Simulation Time: 1 hour and 8 minutes

**References**

ISDB-T Design Examples

2-Layer ISDB-T System Design

DTV_ISDBSystem_prj Design Name
DsnDTV_ISDBTwoLay_System.dsn

Features

- 2-layer system design
- Different modulation mode and code rate for each layer
- Reed-Solomon coding and decoding
- Punctured convolutional coding and decoding
- Mode 1 OFDM adaptation
- Byte-wise interleaving and deinterleaving
- Time interleaving and deinterleaving
- Frequency interleaving and deinterleaving

Description

This design is a 2-layer system example of ISDB-T. It includes 13 segments (5 segments for Layer A, 8 segments for Layer B). Modulation modes and code rates are DQPSK, 1/2, and 64-QAM, 7/8 in Layer A and Layer B, respectively.

In the OFDM adaptation, the 2-layer full system design works in mode 1. The IFFT/FFT size is 2048; the guard interval is 1/8 of IFFT size. The simulation channel is the TDMA channel. The byte in transmitter and receiver, the received DQPSK and 64-QAM constellations are shown in the simulation results.

Schematics

Figure 9-90 shows the schematic for this design; the coding and decoding subnetwork designs are shown in Figure 9-91 through Figure 9-96.
ISDB-T Design Examples

Figure 9-92. sub_ISDBOFDM_TwoLayDemod.dsn

Figure 9-93. sub_ISDBTwoLay_ChCoder.dsn
Figure 9-94. sub_ISDBTwoLay_ChDecoder.dsn

Figure 9-95. sub_ISDBTwoLay_FreqInterlv.dsn
Specifications

<table>
<thead>
<tr>
<th>Specification (Parameter)</th>
<th>Simulation Type</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Carriers</td>
<td>ADS Ptolemy</td>
<td>96</td>
</tr>
<tr>
<td>Segments</td>
<td>ADS Ptolemy</td>
<td>13</td>
</tr>
<tr>
<td>IFFTSize</td>
<td>ADS Ptolemy</td>
<td>2048</td>
</tr>
<tr>
<td>FFTSize</td>
<td>ADS Ptolemy</td>
<td>2048</td>
</tr>
<tr>
<td>Guard</td>
<td>ADS Ptolemy</td>
<td>256</td>
</tr>
<tr>
<td>IFFTOrder</td>
<td>ADS Ptolemy</td>
<td>11</td>
</tr>
<tr>
<td>FFTOrder</td>
<td>ADS Ptolemy</td>
<td>11</td>
</tr>
<tr>
<td>MaxDelay</td>
<td>ADS Ptolemy</td>
<td>2047</td>
</tr>
<tr>
<td>SegmentsA</td>
<td>ADS Ptolemy</td>
<td>5</td>
</tr>
<tr>
<td>SegmentsB</td>
<td>ADS Ptolemy</td>
<td>8</td>
</tr>
<tr>
<td>Ru</td>
<td>ADS Ptolemy</td>
<td>0.9</td>
</tr>
</tbody>
</table>

Notes

1. The propagation channel model used in this example is a standard TDMA channel.

2. The DelayByte parameter of the punctured convolutional decoder model (DTV_PunConvDecoder) is used to adjust delays caused by OFDM demodulation (which is one OFDM symbol) and two additional OFDM symbols.
delay which is corresponding to the bit interleaving procedure because delay adjustment for bit interleaving is 2 OFDM symbols.

3. Total delay caused by byte-wise interleaving and deinterleaving is 17*11*12 bytes (corresponding to 11 TSPs). Because the OFDM symbol is transmitted as an OFDM frame, in this design the delay of the TSPs is adjusted to one OFDM frame.

4. Total delay caused by time interleaving and deinterleaving is adjusted in accordance with an integer I as shown in table 4-3 in ISDB-T. Because I = 4 in this 2-layer design, the number of OFDM frames to be delayed by delay adjustment and time interleaving is 2.

5. The last delay adjusted in this design generates the number of total null TSP bytes which is needed per operation in DTV_SynTwoLayTSP model.

Simulation Results

Figure 9-97 shows the transmitted and received TSP data.

Figure 9-98 and Figure 9-99 show the received constellation of Layer A and B.
### ISDB-T Design Examples

<table>
<thead>
<tr>
<th>Index</th>
<th>Rx Byte</th>
<th>Index</th>
<th>Tx Byte</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>5</td>
<td>6</td>
<td>6</td>
<td>7</td>
</tr>
<tr>
<td>6</td>
<td>7</td>
<td>7</td>
<td>8</td>
</tr>
<tr>
<td>7</td>
<td>8</td>
<td>8</td>
<td>9</td>
</tr>
</tbody>
</table>

**Figure 9-97. Compare RxTSP with TxTSP**

![Constellation Diagram](image)

**Figure 9-98. Received Constellation for Layer A**
Figure 9-99. Received Constellation for Layer B

Benchmark

• Hardware Platform: Pentium II 200 MHz, 128 MB memory
• Software Platform: WindowsNT 4.0 Workstation, ADS 1.3
• Data Points: 188*((12*5+63*8)*3+47+47) Bytes
• Simulation Time: 1 hour and 26 minutes

References

ISDB-T Design Examples

**TMCC in ISDB-T 1-Layer System Design**

**DTV_ISDBSystem_prj Design Name**

DsnDTV_TMCCMod.dsn

**Features**

Displays include:

- TMCCInfoT (102 TMCC information bits in the transmitter)
- TMCCT (204 TMCC bits in the transmitter after channel coding)
- TMCCInfoR (102 TMCC information bits in the receiver after channel decoding)
- TMCCCR (204 TMCC bits in the receiver after TMCC modulation, AWGN channel, and demodulation)

**Description**

This design example illustrates how to use TMCC models in ISDB-T 1-layer systems. It includes TMCC information multiplexing and de-multiplexing, channel coding and decoding, and modulation and demodulation of TMCC carriers.

If the 1-layer ISDB-T system works in mode 3 and uses 13 segments, its modulation scheme of OFDM carrier is DQPSK, coding rate of inner code is 1/2, length of time interleaving is 2, then the first 122 TMCC bits should be:

```
0
0 0 1 1 0 1 1 1 1 0 1 1 1 0
1 1 1
0 0
1 1 1 1
0
0
0 0 0
0 0 0
0 1 0
1 1 0 1
1 1 1
1 1 1
1 1 1
1 1 1
1 1 1
1 1 1
1 1 1
```

9-78   DTV ISDB System
ISDB-T Design Examples

We can compare the results of this design with the previous 122 TMCC bits in the dds file.

Schematics

Figure 9-100 shows the schematic for this design.

Simulation Results

Figure 9-101 shows the 102 TMCC information bits in the transmitter before channel coding and in the receiver after channel decoding. The first column is the information bits in the transmitter; the second column is the information bits in the receiver; the third column is the error between the information bits.
<table>
<thead>
<tr>
<th>Index</th>
<th>$i_{\text{info}1}$</th>
<th>$i_{\text{info}2}$</th>
<th>$\text{err}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>1</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>2</td>
<td>1.000</td>
<td>1.000</td>
<td>0.000</td>
</tr>
<tr>
<td>3</td>
<td>1.000</td>
<td>1.000</td>
<td>0.000</td>
</tr>
<tr>
<td>4</td>
<td>1.000</td>
<td>1.000</td>
<td>0.000</td>
</tr>
<tr>
<td>5</td>
<td>1.000</td>
<td>1.000</td>
<td>0.000</td>
</tr>
<tr>
<td>6</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>7</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>8</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>9</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>10</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>11</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>12</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>13</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>14</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>15</td>
<td>1.000</td>
<td>1.000</td>
<td>0.000</td>
</tr>
<tr>
<td>16</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>17</td>
<td>1.000</td>
<td>1.000</td>
<td>0.000</td>
</tr>
<tr>
<td>18</td>
<td>1.000</td>
<td>1.000</td>
<td>0.000</td>
</tr>
<tr>
<td>19</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>20</td>
<td>1.000</td>
<td>1.000</td>
<td>0.000</td>
</tr>
<tr>
<td>21</td>
<td>1.000</td>
<td>1.000</td>
<td>0.000</td>
</tr>
<tr>
<td>22</td>
<td>1.000</td>
<td>1.000</td>
<td>0.000</td>
</tr>
<tr>
<td>23</td>
<td>1.000</td>
<td>1.000</td>
<td>0.000</td>
</tr>
<tr>
<td>24</td>
<td>1.000</td>
<td>1.000</td>
<td>0.000</td>
</tr>
<tr>
<td>25</td>
<td>1.000</td>
<td>1.000</td>
<td>0.000</td>
</tr>
<tr>
<td>26</td>
<td>1.000</td>
<td>1.000</td>
<td>0.000</td>
</tr>
<tr>
<td>27</td>
<td>1.000</td>
<td>1.000</td>
<td>0.000</td>
</tr>
<tr>
<td>28</td>
<td>1.000</td>
<td>1.000</td>
<td>0.000</td>
</tr>
<tr>
<td>29</td>
<td>1.000</td>
<td>1.000</td>
<td>0.000</td>
</tr>
<tr>
<td>30</td>
<td>1.000</td>
<td>1.000</td>
<td>0.000</td>
</tr>
<tr>
<td>31</td>
<td>1.000</td>
<td>1.000</td>
<td>0.000</td>
</tr>
<tr>
<td>32</td>
<td>1.000</td>
<td>1.000</td>
<td>0.000</td>
</tr>
</tbody>
</table>

Figure 9-101. 102 TMCC information Bits in the Transmitter and Receiver

Figure 9-102 shows the 204 TMCC bits in the transmitter before modulation and in the receiver after demodulation. The first column is the transmitter bits; the second column is the receiver bits; the third column is the error between the information bits.
### ISDB-T Design Examples

**Figure 9-102. 204 TMCC bits in the Transmitter and Receiver**

<table>
<thead>
<tr>
<th>Index</th>
<th>TMCC T</th>
<th>TMCC R</th>
<th>err T</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>1</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>3</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>4</td>
<td>0.000</td>
<td>1.000</td>
<td>0.000</td>
</tr>
<tr>
<td>6</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>7</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>8</td>
<td>0.000</td>
<td>1.000</td>
<td>0.000</td>
</tr>
<tr>
<td>9</td>
<td>1.000</td>
<td>1.000</td>
<td>0.000</td>
</tr>
<tr>
<td>10</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>11</td>
<td>1.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>12</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>13</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>14</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>15</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>16</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>17</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>18</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>19</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>20</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>21</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>22</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>23</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>24</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>25</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>26</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>27</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>28</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>29</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>30</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>31</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>32</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
</tbody>
</table>

**References**

TMCC in ISDB-T 3-Layer System Design

DTV_ISDBSystem_prj Design Name
DsnDTV_TMCCThrLay.dsn

Features
Displays include:

• TMCCIfoT (102 TMCC information bits in the transmitter)
• TMCCr 204 TMCC bits in the transmitter after channel coding)
• TMCCIfoR (102 TMCC information bits in the receiver after channel decoding)
• TMCCr (204 TMCC bits in the receiver after TMCC modulation, AWGN channel, and demodulation)

Description
This design example illustrates how to use TMCC models in ISDB-T 3-layer systems. It includes TMCC information multiplexing and de-multiplexing, channel coding and decoding, and modulation and demodulation of TMCC carriers.

Schematics
Figure 9-103 shows the schematic for this design.
Figure 9-104 lists the 102 bits TMCC information bits in the transmitter before channel coding and in the receiver after channel decoding. The first column is the 102 TMCC information bits in the transmitter; the second column is the 102 information bits in the receiver; the third column is the error between both 102 TMCC information bits.

Figure 9-105 shows the 204 bits TMCC bits in the transmitter before modulator and in the receiver after de-modulator. The first column is the 204 TMCC bits in the transmitter, the second column is the 204 TMCC bits in the receiver, and the third column is the error between both 204 TMCC information bits.
<table>
<thead>
<tr>
<th>Index</th>
<th>TMCC InfoT</th>
<th>TMCC InfoR</th>
<th>err</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>1</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>2</td>
<td>1.000</td>
<td>1.000</td>
<td>0.000</td>
</tr>
<tr>
<td>3</td>
<td>1.000</td>
<td>1.000</td>
<td>0.000</td>
</tr>
<tr>
<td>4</td>
<td>1.000</td>
<td>1.000</td>
<td>0.000</td>
</tr>
<tr>
<td>5</td>
<td>1.000</td>
<td>1.000</td>
<td>0.000</td>
</tr>
<tr>
<td>6</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>7</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>8</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>9</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>10</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>11</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>12</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>13</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>14</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>15</td>
<td>1.000</td>
<td>1.000</td>
<td>0.000</td>
</tr>
<tr>
<td>16</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>17</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>18</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>19</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>20</td>
<td>1.000</td>
<td>1.000</td>
<td>0.000</td>
</tr>
<tr>
<td>21</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>22</td>
<td>1.000</td>
<td>1.000</td>
<td>0.000</td>
</tr>
<tr>
<td>23</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>24</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>25</td>
<td>1.000</td>
<td>1.000</td>
<td>0.000</td>
</tr>
<tr>
<td>26</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>27</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>28</td>
<td>1.000</td>
<td>1.000</td>
<td>0.000</td>
</tr>
<tr>
<td>29</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>30</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>31</td>
<td>1.000</td>
<td>1.000</td>
<td>0.000</td>
</tr>
<tr>
<td>32</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>33</td>
<td>1.000</td>
<td>1.000</td>
<td>0.000</td>
</tr>
<tr>
<td>34</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>35</td>
<td>1.000</td>
<td>1.000</td>
<td>0.000</td>
</tr>
<tr>
<td>36</td>
<td>1.000</td>
<td>1.000</td>
<td>0.000</td>
</tr>
<tr>
<td>37</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>38</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
</tbody>
</table>

Figure 9-104. 102 Bits TMCC Information in Transmitter and Receiver
ISDB-T Design Examples

Figure 9-105. 204 TMCC Bits in Transmitter and Receiver

Benchmark

- Hardware Platform: Pentium III 450 MHz, 512 MB memory
- Software Platform: Windows NT 4.0 Workstation, ADS 1.3
- Data Points: 408
- Simulation Time: 5 seconds

References

Chapter 10: DVB-T Receiver Design
Examples
DVB-T Receiver Design Examples

DVB-T AWGN Performance

DTV_DVBT_Rx_AWGN_Option1.dsn
DTV_DVBT_Rx_AWGN_Option2.dsn

Features: Option 1 Design

- 2k mode
- 64-QAM mapping
- 2/3 coding rate
- 1/32 guard interval
- Signal_Generation_VARs and RF_Channel_Measurement_VARs parameters can be changed by the user

Features: Option 2 Design

- 2k mode
- 16-QAM mapping
- 3/4 coding rate
- 1/32 guard interval
- Signal_Generation_VARs and RF_Channel_Measurement_VARs parameters can be changed by the user

Description

These designs measure the performance of a DVB-T receiver with AWGN. BER is measured after Viterbi decoding.

The schematic is shown in Figure 10-1.
Figure 10-1. DTV_DVBT_Rx_AWGN Schematic
Simulation Results

For Option 1, an 18.3 dB carrier-to-noise ratio is used. Simulation results displayed in DTV_DVBT_Rx_AWGN_Option1.dds are shown in Figure 10-2. The demodulated constellations are shown in Figure 10-3.

<table>
<thead>
<tr>
<th>Index</th>
<th>BER</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000</td>
<td>2.995E-4</td>
</tr>
</tbody>
</table>

Figure 10-2. BER for Option 1

For Option 2, a 13.8 dB carrier-to-noise ratio is used. Simulation results displayed in DTV_DVBT_Rx_AWGN_Option2.dds are shown in Figure 10-4. The demodulated constellation is shown in Figure 10-5.

<table>
<thead>
<tr>
<th>Index</th>
<th>BER</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000</td>
<td>2.48E-4</td>
</tr>
</tbody>
</table>

Figure 10-4. BER for Option 2
Figure 10-5. Constellation for Option 2

Benchmark

- Hardware Platform: Pentium IV 2.26 GHz, 512 MB memory
- Software Platform: Windows 2000, ADS 2003C
- Simulation Time: approximately 1 minute

References

DVB-T Receiver Design Examples

DVB-T Digital Adjacent Channel Performance

DTV_DVBT_Rx_Digital_Adj_Option1.dsn
DTV_DVBT_Rx_Digital_Adj_Option2.dsn

Features: Option 1 Design

• 2k mode
• 64-QAM mapping
• 2/3 coding rate
• 1/32 guard interval
• Signal.Generation.VARs and Measurement.VARs parameters can be changed by the user

Features: Option 2 Design

• 2k mode
• 16-QAM mapping
• 3/4 coding rate
• 1/32 guard interval
• Signal.Generation.VARs and Measurement.VARs parameters can be changed by the user

Description

These designs measure the performance of a DVB-T receiver with a digital adjacent channel. BER is measured after Viterbi decoding.

The schematic is shown in Figure 10-6.
Figure 10-6. DTV_DVBT_Rx_Digital_Adj Schematic
Simulation Results

Option 1 uses a 30 dB adjacent channel signal interference level relative to the wanted DVB-T signal.

Option 2 uses a 32 dB adjacent channel signal interference level relative to the wanted DVB-T signal.

Simulation results displayed in DTV_DVBT_Rx_Digital_Adj_Option1.dds and DTV_DVBT_Rx_Digital_Adj_Option2.dds are shown in Figure 10-7.

<table>
<thead>
<tr>
<th>Index</th>
<th>BER</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000</td>
<td>0.000</td>
</tr>
</tbody>
</table>

Figure 10-7. BER for Option 1 and Option 2
The spectrum for Option 1 is shown in Figure 10-8; the demodulated constellation is shown in Figure 10-9.
The spectrum for Option 2 is shown in Figure 10-10; the demodulated constellation is shown in Figure 10-11.
Figure 10-11. Constellation for Option 2
DVB-T Receiver Design Examples

**Benchmark**

- Hardware Platform: Pentium IV 2.26 GHz, 512 MB memory
- Software Platform: Windows 2000, ADS 2003C
- Simulation Time: approximately 5 minutes

**References**


DVB-T Impulse Interference Performance

DTV_DVBT_Rx_Impulse_Option1.dsn
DTV_DVBT_Rx_Impulse_Option2.dsn

Features: Option 1 Design

• 2k mode
• 64-QAM mapping
• 2/3 coding rate
• 1/32 guard interval
• Signal_Generation_VARs and RF_Channel_Measurement_VARs parameters can be changed by the user.

Features: Option 2 Design

• 2k mode
• 16-QAM mapping
• 3/4 coding rate
• 1/32 guard interval
• Signal_Generation_VARs and RF_Channel_Measurement_VARs parameters can be changed by the user.

Description

These designs measure the performance of a DVB-T receiver with impulse interference. BER is measured after Viterbi decoding.

The impulse interference is made up of bursts that are generated by gating a Gaussian noise source. As illustrated in Figure 10-12, separation of bursts is fixed at 10 msec and pulse duration is fixed at 250 nsec.
These design supports the tests listed in Table 10-1. The tolerance of the receiver to the test signal should exceed its tolerance to ungated Gaussian noise by a tolerance factor that is generally expressed in dB. The tolerance factor is independent of modulation mode, receiver implementation margin, and degradation criterion.

<table>
<thead>
<tr>
<th>Test_No</th>
<th>Pulses per Burst</th>
<th>Pulse Spacing (µsec)</th>
<th>Burst Duration (µsec)</th>
<th>Tolerance Factor (dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0.25</td>
<td>0.25</td>
<td>29.5</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>20</td>
<td>40</td>
<td>26.5</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>20</td>
<td>80</td>
<td>23.5</td>
</tr>
<tr>
<td>4</td>
<td>12</td>
<td>12</td>
<td>144</td>
<td>18.7</td>
</tr>
<tr>
<td>5</td>
<td>20</td>
<td>1.5</td>
<td>30</td>
<td>16.5</td>
</tr>
<tr>
<td>6</td>
<td>40</td>
<td>0.75</td>
<td>30</td>
<td>13.5</td>
</tr>
</tbody>
</table>

The schematic is shown in Figure 10-13.
Figure 10-13. DTV_DVBT_Rx_Impulse Schematic
Simulation Results

Option 1 uses a 16.5 dB carrier-to-noise ratio and a tolerance factor of 23.5 dB. Option 2 uses a 12.5 dB carrier-to-noise ratio and a tolerance factor of 23.5 dB. Simulation results displayed in DTV_DVBT_Rx_Impulse_Option1.dds and DTV_DVBT_Rx_Impulse_Option2.dds are shown in Figure 10-14.

<table>
<thead>
<tr>
<th>Index</th>
<th>BER</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000</td>
<td>0.000</td>
</tr>
</tbody>
</table>

Figure 10-14. BER for Option 1 and Option 2

The demodulated constellations are shown in Figure 10-15 and Figure 10-16.

Figure 10-15. Constellation for Option 1
Figure 10-16. Constellation for Option 2

Benchmark

- Hardware Platform: Pentium IV 2.26 GHz, 512 MB memory
- Software Platform: Windows 2000, ADS 2003C
- Simulation Time: approximately 2 minutes

References


DVB-T Receiver Design Examples

DVB-T Long Delay Channel Performance

DTV_DVBT_Rx_LongDelay_Option1.dsn
DTV_DVBT_Rx_LongDelay_Option2.dsn

Features: Option 1 Design

- 2k mode
- 64-QAM mapping
- 2/3 coding rate
- 1/32 guard interval
- Signal_Generation_VARs and RF_Channel_Measurement_VARs parameters can be set by the user

Features: Option 2 Design

- 2k mode
- 16-QAM mapping
- 3/4 coding rate
- 1/32 guard interval
- Signal_Generation_VARs and RF_Channel_Measurement_VARs parameters can be set by the user

Description

These designs measure the performance of a DVB-T receiver with a long delay channel. The long delay combines 6 echoes. There is a dominant direct path and 5 echoes and one outside the guard interval. Path delay and attenuation of the long delay multipath channel are [0, 5, 14, 35, 54, 75] µsec and [0, 9, 22, 25, 27, 28] dB. The phase is set to 0 at channel center.

BER is measured after Viterbi decoding.

The schematic is shown in Figure 10-17.
Simulation Results

Option 1 uses a 22.3 dB carrier-to-noise ratio. Simulation results displayed in DTV_DVBT_Rx_LongDelay_Option1.dds are shown in Figure 10-18. The demodulated constellations are shown in Figure 10-19.

<table>
<thead>
<tr>
<th>Index</th>
<th>BER</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000</td>
<td>2.13E-4</td>
</tr>
</tbody>
</table>

Figure 10-18. BER for Option 1
Option 2 uses a 15.8 dB carrier-to-noise ratio. Simulation results displayed in DTV_DVBT_Rx_LongDelay_Option2.dds are shown in Figure 10-20. The demodulated constellations are shown in Figure 10-21.

<table>
<thead>
<tr>
<th>index</th>
<th>BER</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000</td>
<td>3.47E-4</td>
</tr>
</tbody>
</table>

Figure 10-20. BER for Option 2
Figure 10-21. Constellation for Option 2

Benchmark

- Hardware Platform: Pentium IV 2.26 GHz, 512 MB memory
- Software Platform: Windows 2000, ADS 2003C
- Simulation Time: approximately 15 minutes for Option 1 and 30 minutes for Option 2

References

DVB-T Receiver Design Examples

DVB-T Analog Adjacent Channel Performance

DTV_DVBT_Rx_PAL_Adj_Option1.dsn
DTV_DVBT_Rx_PAL_Adj_Option2.dsn

Features: Option 1 Design

• 2k mode
• 64-QAM mapping
• 2/3 coding rate
• 1/32 guard interval
• Signal_Generation_VARs and Measurement_VARs parameters can be set by the user

Features: Option 2 Design

• 2k mode
• 16-QAM mapping
• 3/4 coding rate
• 1/32 guard interval
• Signal_Generation_VARs and Measurement_VARs parameters can be set by the user
Description

These designs measure the protection ratio of an adjacent channel PAL signal. BER is measured after Viterbi decoding.

The schematic is shown in Figure 10-22.
Simulation Results

For Option 1, the adjacent channel PAL signal interference level relative to the wanted DVB-T signal is 35 dB.

For Option 2, the adjacent channel PAL signal interference level relative to the wanted DVB-T signal is 37 dB.

Simulation results displayed in DTV_DVBT_Rx_PAL_Adj_Option1.dds and DTV_DVBT_Rx_PAL_Adj_Option2.dds are shown in Figure 10-23.

<table>
<thead>
<tr>
<th>Index</th>
<th>BER</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000</td>
<td>0.003</td>
</tr>
</tbody>
</table>

Figure 10-23. BER for Option 1 and Option 2
For Option 1, the spectrum is shown in Figure 10-24; the demodulated constellation is shown in Figure 10-25.

Figure 10-24. Spectrum for Option 1

Figure 10-25. Constellation for Option 1
For Option 2, the spectrum is shown in Figure 10-26; the demodulated constellation is shown in Figure 10-27.
Benchmark

- Hardware Platform: Pentium IV 2.26 GHz, 512 MB memory
- Software Platform: Windows 2000, ADS 2003C
- Simulation Time: approximately 10 minutes

References


DVB-T Analog Co-Channel Performance

DTV_DVBT_Rx_PAL_Cochannel_Option1.dsn
DTV_DVBT_Rx_PAL_Cochannel_Option2.dsn

Features: Option 1 Design

- 2k mode
- 64-QAM mapping
- 2/3 coding rate
- 1/32 guard interval
- Signal_Generation_VARs and Measurement_VARs parameters can be set by the user

Features: Option 2 Design

- 2k mode
- 16-QAM mapping
- 3/4 coding rate
- 1/32 guard interval
- Signal_Generation_VARs and Measurement_VARs parameters can be set by the user

Description

These designs measure the protection ratio of a co-channel PAL signal. BER is measured after Viterbi decoding.

The schematic is shown in Figure 10-28.
Simulation Results

For Option 1, the co-channel PAL signal interference level relative to the wanted DVB-T signal is 0 dB. Simulation results displayed in DTV_DVBT_Rx_PAL_Cochannel_Option1.dds are shown in Figure 10-29.

For Option 2, the co-channel PAL signal interference level relative to the wanted DVB-T signal is 0 dB. Simulation results displayed in DTV_DVBT_Rx_PAL_Cochannel_Option2.dds are shown in Figure 10-30.

<table>
<thead>
<tr>
<th>Index</th>
<th>BER</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000</td>
<td>2.852E-4</td>
</tr>
</tbody>
</table>

Figure 10-29. BER for Option 1

<table>
<thead>
<tr>
<th>Index</th>
<th>BER</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000</td>
<td>8.160E-4</td>
</tr>
</tbody>
</table>

Figure 10-30. BER for Option 2
For Option 1, the spectrum is shown in Figure 10-31; the demodulated constellation is shown in Figure 10-32. The PAL source peak power shown in Figure 10-33 is -59.910 dBm, which is the same as the schematic setting.
Figure 10-33. PAL Source Peak Power for Option 1
DVB-T Receiver Design Examples

For Option 2, the demodulated constellation is shown in Figure 10-34; the demodulated constellation is shown in Figure 10-35. The PAL source peak power shown in Figure 10-36 is -59.910 dBm, which is the same as the schematic setting.

![Figure 10-34. Spectrum for Option 2](image)

![Figure 10-35. Constellation for Option 2](image)
Figure 10-36. PAL Source Peak Power for Option 2
DVB-T Receiver Design Examples

Benchmark

- Hardware Platform: Pentium IV 2.26 GHz, 512 MB memory
- Software Platform: Windows 2000, ADS 2003C
- Simulation Time: approximately 5 minutes

References


DVB-T Image Channel Performance

DTV_DVBT_Rx_PAL_Image_Option1.dsn
DTV_DVBT_Rx_PAL_Image_Option2.dsn

Features: Option 1 Design
- 2k mode
- 64-QAM mapping
- 2/3 coding rate
- 1/32 guard interval
- Signal_Generation_VARS and Measurement_VARS parameters can be set by the user

Features: Option 2 Design
- 2k mode
- 16-QAM mapping
- Coding rate is 3/4
- Guard interval is 1/32
- Signal_Generation_VARS and Measurement_VARS parameters can be set by the user

Description
These designs measure the protection ratio of an image channel PAL signal.
The schematic is shown in Figure 10-37.
Simulation Results

BER is measured after Viterbi decoding.

For Option 1, the image channel PAL signal interference level relative to wanted DVB-T signal is 46 dB.

For Option 2, the image channel PAL signal interference level relative to wanted DVB-T signal is 50 dB.

Simulation results displayed in DTV_DVBT_Rx_PAL_Image_Option1.dds and DTV_DVBT_Rx_PAL_Image_Option2.dds are shown in Figure 10-38.

<table>
<thead>
<tr>
<th>Index</th>
<th>BER</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000</td>
<td>0.000</td>
</tr>
</tbody>
</table>

Figure 10-38. BER for Option 1 and Option 2
For Option 1, the demodulated constellation is shown in Figure 10-39.

Figure 10-39. Constellation for Option 1

For Option 2, the demodulated constellation is shown in Figure 10-40.

Figure 10-40. Constellation for Option 2
DVB-T Receiver Design Examples

Benchmark

- Hardware Platform: Pentium IV 2.26 GHz, 512 MB memory
- Software Platform: Windows 2000, ADS 2003C
- Simulation Time: approximately 6 hours for Option 1 and 9 hours for Option 2

References


DVB-T Rayleigh Channel (P1) Performance

DTV_DVBT_Rx_Rayleigh_Option1.dsn
DTV_DVBT_Rx_Rayleigh_Option2.dsn

Features: Option 1 Design

• 2k mode
• 64-QAM mapping
• 2/3 coding rate
• 1/32 guard interval
• Signal.Generation.VARs and RF_Channel.Measurement.VARs parameters can be changed by the user

Features: Option 2 Design

• 2k mode
• 16-QAM mapping
• 3/4 coding rate
• 1/32 guard interval
• Signal.Generation.VARs and RF_Channel.Measurement.VARs parameters can be changed by the user

Description

These designs measure the performance of a DVB-T receiver with a Rayleigh channel (P1).

BER is measured after Viterbi decoding.

The schematic is shown in Figure 10-41.
For Option 1, a 22.6 dB carrier-to-noise ratio is used. Simulation results displayed in DTV_DVBT_Rx_Rayleigh_Option1.dds are shown in Figure 10-42. The demodulated constellation is shown in Figure 10-43.

Figure 10-42. BER for Option 1
Figure 10-43. Constellation for Option 1
For Option 2, a 19.1 dB carrier-to-noise ratio is used. Simulation results displayed in DTV_DVBT_Rx_Rayleigh_Option2.dds are shown in Figure 10-44. The demodulated constellation is shown in Figure 10-45.

### Benchmark

- Hardware Platform: Pentium IV 2.26 GHz, 512 MB memory
- Software Platform: Windows 2000, ADS 2003C
- Simulation Time: approximately 1 minute
References


DVB-T Ricean Channel (F1) Performance

DTV_DVBT_Rx_Ricean_Option1.dsn
DTV_DVBT_Rx_Ricean_Option2.dsn

Features: Option 1 Design
- 2k mode
- 64-QAM mapping
- 2/3 coding rate
- 1/32 guard interval
- Signal_Generation_VARs and RF_Channel_Measurement_VARs parameters can be changed by the user

Features: Option 2 Design
- 2k mode
- 16-QAM mapping
- 3/4 coding rate
- 1/32 guard interval
- Signal_Generation_VARs and RF_Channel_Measurement_VARs parameters can be changed by the user

Description
These designs measure the performance of a DVB-T receiver with a Ricean channel (F1). BER is measured after Viterbi decoding.

The schematic is shown in Figure 10-46.
Figure 10-46. DTV_DVBT_Rx_Ricean Schematic
Simulation Results

For option 1, a 19.4 dB carrier-to-noise ratio is used. Simulation results displayed in DTV_DVBT_Rx_Ricean_Option1.dds are shown in Figure 10-47. The demodulated constellation is shown in Figure 10-48.

<table>
<thead>
<tr>
<th>Index</th>
<th>BER</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000</td>
<td>2.20E-4</td>
</tr>
</tbody>
</table>

Figure 10-47. BER for Option 1

Figure 10-48. Constellation for Option 1
For option 2, a 14.9 dB carrier-to-noise ratio is used. Simulation results displayed in DTV_DVBT_Rx_Ricean_Option2.dds are shown in Figure 10-49. The demodulated constellation is shown in Figure 10-50.

<table>
<thead>
<tr>
<th>Index</th>
<th>BER</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000</td>
<td>2.67E-4</td>
</tr>
</tbody>
</table>

Figure 10-49. BER for Option 2

Figure 10-50. Constellation for Option 2

Benchmark

- Hardware Platform: Pentium IV 2.26 GHz, 512 MB memory
- Software Platform: Windows 2000, ADS 2003C
- Simulation Time: approximately 1 minute

References

DVB-T Receiver Design Examples

DVB-T Receiver Minimum Input Level Sensitivity

DTV_DVBT_Rx_Sensitivity_Option1.dsn
DTV_DVBT_Rx_Sensitivity_Option2.dsn

Features: Option 1 Design
- 2k mode
- 64-QAM mapping
- 2/3 coding rate
- 1/32 guard interval
- Signal_Generation_VARs and Measurement_VARs parameters can be changed by the user

Features: Option 2 Design
- 2k mode
- 16-QAM mapping
- 3/4 coding rate
- 1/32 guard interval
- Signal_Generation_VARs and Measurement_VARs parameters can be changed by the user

Description
These designs measure the performance of a DVB-T receiver with minimum input level sensitivity. BER is measured after Viterbi decoding.

The schematic is shown in Figure 10-51.
Figure 10-51. DTV_DVBT_Rx_Sensitivity Schematic
Simulation Results

BER is measured after Viterbi decoding.

Option 1 uses a signal power level of -77.9 dBm. Simulation results displayed in DTV_DVBT_Rx_Sensitivity_Option1.dds are shown in Figure 10-52. The demodulated constellation is shown in Figure 10-53.

<table>
<thead>
<tr>
<th>Index</th>
<th>BER</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000</td>
<td>1.36E-6</td>
</tr>
</tbody>
</table>

Figure 10-52. BER for Option 1

Figure 10-53. Constellation for Option 1
Option 2 uses a signal power level of -82.1 dBm. Simulation results displayed in DTV_DVBT_Rx_Sensitivity_Option2.dds are shown in Figure 10-54. The demodulated constellation is shown in Figure 10-55.

<table>
<thead>
<tr>
<th>Index</th>
<th>BER</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000</td>
<td>7.51E-10</td>
</tr>
</tbody>
</table>

Figure 10-54. BER for Option 2

Figure 10-55. Constellation for Option 2
DVB-T Receiver Design Examples

Benchmark

• Hardware Platform: Pentium IV 2.26 GHz, 512 MB memory
• Software Platform: Windows 2000, ADS 2003C
• Simulation Time: approximately 1 minute

References

DVB-T Short Delay Channel Performance

DTV_DVBT_Rx_ShortDelay_Option1.dsn
DTV_DVBT_Rx_ShortDelay_Option2.dsn

Features: Option 1 Design
- 2k mode
- 64-QAM mapping
- 2/3 coding rate
- 1/32 guard interval
- Signal_Generation_VARs and RF_Channel_Measurement_VARs parameters can be changed by the user

Features: Option 2 Design
- 2k mode
- 16-QAM mapping
- 3/4 coding rate
- 1/32 guard interval
- Signal_Generation_VARs and RF_Channel_Measurement_VARs parameters can be changed by the user

Description
These designs measure the performance of a DVB-T receiver with a short delay channel. The short delay combines 6 echoes. All echoes are within the guard interval and none is dominant: there is no direct path. Path delay and attenuation of the short delay multipath channel are [0, 0.05, 0.4, 1.45, 2.3, 2.8] µsec and [2.8, 0.0, 3.8, 0.1, 2.6, 1.3] dB. The phase is set to 0 at the channel center.

BER is measured after Viterbi decoding.

The schematic is shown in Figure 10-56.
Figure 10-56. DTV_DVBT_Rx_ShortDelay Schematic
Simulation Results

Option 1 has a carrier-to-noise ratio of 22.3 dB. Simulation results displayed in DTV_DVBT_Rx_ShortDelay_Option1.dds are shown in Figure 10-57. The demodulated constellation is shown in Figure 10-58.

<table>
<thead>
<tr>
<th>Index</th>
<th>BER</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000</td>
<td>1.42E-4</td>
</tr>
</tbody>
</table>

Figure 10-57. BER for Option 1

Figure 10-58. Constellation for Option 1
Option 2 has a carrier-to-noise ratio of 18.8 dB. Simulation results displayed in DTV_DVBT_Rx_ShortDelay_Option2.dds are shown in Figure 10-59. The demodulated constellation is shown in Figure 10-60.

<table>
<thead>
<tr>
<th>Index</th>
<th>BER</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000</td>
<td>1.0E-4</td>
</tr>
</tbody>
</table>

**Figure 10-59. BER for Option 2**

**Figure 10-60. Constellation for Option 2**

**Benchmark**

- Hardware Platform: Pentium IV 2.26 GHz, 512 MB memory
- Software Platform: Windows 2000, ADS 2003C
- Simulation Time: approximately 15 minutes for Option 1 and 25 minutes for Option 2

**References**


DVB-T Single Echo Channel Performance

DTV_DVBT_Rx_SingleEcho_Option1.dsn
DTV_DVBT_Rx_SingleEcho_Option2.dsn

Features: Option 1 Design
• 2k mode
• 64-QAM mapping
• 2/3 coding rate
• 1/32 guard interval
• Signal_Generation_VARs and RF_Channel_Measurement_VARs parameters can be changed by the user

Features: Option 2 Design
• 2k mode
• 16-QAM mapping
• 3/4 coding rate
• 1/32 guard interval
• Signal_Generation_VARs and RF_Channel_Measurement_VARs parameters can be changed by the user

Description
These designs measure the performance of a DVB-T receiver with a single echo channel. Path delay of the echo channel is 3.5 µsec and the attenuation is 0 dB. The echo phase is set to 90 at the channel center.

BER is measured after Viterbi decoding.

The schematic is shown in Figure 10-61.
Simulation Results

For option 1, a carrier-to-noise ratio of 23.1 dB is used. Simulation results displayed in DTV_DVBT_Rx_SingleEcho_Option1.dds are shown in Figure 10-62. The demodulated constellation is shown in Figure 10-63.

<table>
<thead>
<tr>
<th>Index</th>
<th>BER</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000</td>
<td>1.937E-4</td>
</tr>
</tbody>
</table>

Figure 10-62. BER for Option 1

Figure 10-63. Constellation for Option 1
For option 2, a carrier-to-noise ratio of 21.3 dB is used. Simulation results displayed in DTV_DVBT_Rx_SingleEcho_Option2.dds are shown in Figure 10-64. The demodulated constellation is shown in Figure 10-65.

<table>
<thead>
<tr>
<th>Index</th>
<th>BER</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000</td>
<td>2.0E-04</td>
</tr>
</tbody>
</table>

Benchmark

- Hardware Platform: Pentium IV 2.26 GHz, 512 MB memory
- Software Platform: Windows 2000, ADS 2003C
- Simulation Time: approximately 2 minutes

References


Index

D
DTV_AddFixPhase, 6-2
DTV_BCHCoder, 2-2
DTV_BCHDecoder, 2-5
DTV_BER, 7-2
DTV_CarrierRotator, 4-7
DTV_CarrierScrambler, 4-9
DTV_CDSCDecoder, 4-2
DTV_ChEstimator, 4-14
DTV_CIRNorm, 3-2
DTV_CommCtrl2, 5-2
DTV_CommCtrl3, 5-3
DTV_ConvCoder, 2-10
DTV_ConvCoder_1_2, 2-12
DTV_ConvDecoder, 2-15
DTV_ConvDecoder_1_2, 2-18
DTV_Demapper, 2-20
DTV_DemuxCohSegs, 4-20
DTV_DemuxDiffSegs, 4-26
DTV_DemuxTMCC, 4-33
DTV_DistCtrl2, 5-4
DTV_DistCtrl3, 5-5
DTV_DQPSKDecoder, 2-23
DTV_DQPSKDecoder, 2-25
DTV_DVBT2DChEstimator, 3-13
DTV_DVBT2BitBlockInterlv, 3-4
DTV_DVBTChannel, 3-6
DTV_DVBTChEstimator, 3-9
DTV_DVBDemuxOFDMSym, 3-17
DTV_DVBLoadFFTBuff, 3-21
DTV_DVBMuxOFDMSym, 3-23
DTV_DVBSymDeinterlv2b, 3-27
DTV_DVBSymDeinterlv4b, 3-29
DTV_DVBSymDeinterlv6b, 3-31
DTV_DVBSymInterlv2b, 3-33
DTV_DVBSymInterlv4b, 3-37
DTV_DVBSymInterlv6b, 3-39
DTV_DVBSymInterlvC, 3-41
DTV_DVBTHierInnerDeinterlv, 3-43
DTV_DVBTHierInnerInterlv, 3-45
DTV_DVBTHierReceiver, 3-47
DTV_DVBTHierReceiver_RF, 3-55
DTV_DVBTHierSignalSrc, 3-65
DTV_DVBTHierSignalSrc_RF, 3-70
DTV_DVBTImpulseNoise, 3-76
DTV_DVBTInnerDeinterlv, 3-79
DTV_DVBTInnerInterlv, 3-81
DTV_DVBT_OFDMDemod, 3-83
DTV_DVBT_OFDMMod, 3-88
DTV_DVBTPS, 3-93
DTV_DVBTPSDemod, 3-97
DTV_DVBTPSMod, 3-99
DTV_DVBTRceiver, 3-101
DTV_DVBTRceiver_RF, 3-107
DTV_DVBTSignalSrc, 3-114
DTV_DVBTSignalSrc_RF, 3-118
DTV_DVBTTPSgen, 3-124
DTV_DVBTTPSsource, 3-130
DTV_InsertGuard, 6-4
DTV_IntervalFloat, 2-29
DTV_IntervalInt, 2-31
DTV_InterSegInterlv, 4-35
DTV_LFSRCoder, 4-38
DTV_LoadFFTBuff, 6-6
DTV_LoadFFTBuff, 4-41
DTV_Mapper, 2-34
DTV_M sender, 6-10
DTV_MuxCohSegs, 4-43
DTV_MuxDiffSegs, 4-49
DTV_OFDMEqualizer, 6-13
DTV_PacketC, 4-56
DTV_PacketTMCC, 3-65
DTV_PALSource, 3-114
DTV_PALSource_RF, 3-118
DTV_PNreset, 2-41
DTV_PowerMeasure, 7-4
DTV_PuncCoder, 2-43
DTV_PuncConvCoder, 2-45
DTV_PuncConvDecoder, 2-47
DTV_PuncDecoder, 2-50
DTV_QAM16Coder, 2-52
DTV_QAM16Decoder, 2-56
DTV_QAM64Coder, 2-58
DTV_QAM64Decoder, 2-62
DTV_QPSKDecoder, 2-64
DTV_QPSKDecoder, 2-66
DTV_RemovePhase, 6-15
DTV_RS_Coder, 2-67
DTV_RSDecoder, 2-71
DTV_ScrambleByte, 2-78
DTV_SplitThreeLayData, 5-6
DTV_SplitThreeLayTSP, 5-8