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.


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 TDSCDMA Design Library
   Introduction ............................................................................................................... 1-1  
   Physical Layer ........................................................................................................ 1-1  
   Component Libraries ............................................................................................ 1-3  
      Fully-Coded Source Components ...................................................................... 1-3  
      Measurements Components .............................................................................. 1-4  
      Modems Components ........................................................................................ 1-4  
      Multiplexing and Coding Components ................................................................ 1-5  
      Physical Channels Components ....................................................................... 1-5  
   Receivers ............................................................................................................ 1-6  
   Signal Sources ................................................................................................... 1-6  
   Glossary of Terms ............................................................................................... 1-6  

2 Fully Coded Sources
   TDSCDMA_RefCh..................................................................................................... 2-2  
   TDSCDMA_RefCh_RF............................................................................................. 2-13  

3 Measurements
   TDSCDMA_BER ...................................................................................................... 3-2  
   TDSCDMA_Constellation ......................................................................................... 3-3  
   TDSCDMA_EVM ..................................................................................................... 3-6  
   TDSCDMA_FrameSync............................................................................................ 3-13  
   TDSCDMA_FwdChannel.......................................................................................... 3-16  
   TDSCDMA_RF_CCDF............................................................................................. 3-18  
   TDSCDMA_RF_PwrMeasure................................................................................... 3-20  
   TDSCDMA_RevChannel.......................................................................................... 3-23  

4 Modems
   TDSCDMA_BurstDeMux.......................................................................................... 4-2  
   TDSCDMA_BurstMux............................................................................................. 4-4  
   TDSCDMA_DPCH_DataDeMux............................................................................... 4-6  
   TDSCDMA_DPCH_DataMux................................................................................... 4-14  
   TDSCDMA_DPCH_Mux........................................................................................... 4-20  
   TDSCDMA_Demodulator......................................................................................... 4-23  
   TDSCDMA_Midamble.............................................................................................. 4-25  
   TDSCDMA_Modulator............................................................................................. 4-30  
   TDSCDMA_OnePhyCh............................................................................................ 4-33  
   TDSCDMA_OnePhyChDeMux................................................................................. 4-35  
   TDSCDMA_OVSF.................................................................................................... 4-37  
   TDSCDMA_PSCCH_DataMux.................................................................................. 4-40  
   TDSCDMA_Scramble............................................................................................. 4-42  

5 Multiplexing and Coding Components

TDSCDMA_Sync ...................................................................................................... 4-44
TDSCDMA_1stDeIntlvr .......................................................................................... 5-2
TDSCDMA_1stIntlvr ............................................................................................ 5-4
TDSCDMA_2ndDeIntlvr ...................................................................................... 5-6
TDSCDMA_2ndIntlvr ........................................................................................... 5-8
TDSCDMA_BitScrambling .................................................................................. 5-10
TDSCDMA_CRC_Decoder .................................................................................... 5-12
TDSCDMA_CRC_Encoder .................................................................................... 5-15
TDSCDMA_ChCoding ......................................................................................... 5-18
TDSCDMA_ChDecoding ....................................................................................... 5-21
TDSCDMA_CodeBlkSeg ....................................................................................... 5-25
TDSCDMA_DeCodeBlkSeg .................................................................................. 5-27
TDSCDMA_DePhyChMap ................................................................................... 5-30
TDSCDMA_DePhyChSeg .................................................................................... 5-32
TDSCDMA_DeRadioEqual .................................................................................. 5-36
TDSCDMA_DeRadioSeg ..................................................................................... 5-38
TDSCDMA_DeRateMatch ................................................................................... 5-41
TDSCDMA_DeSubFrameSeg .............................................................................. 5-45
TDSCDMA_PhychMap ....................................................................................... 5-47
TDSCDMA_PhychSeg ........................................................................................ 5-49
TDSCDMA_RadioEqual ....................................................................................... 5-54
TDSCDMA_RadioSeg .......................................................................................... 5-56
TDSCDMA_RateMatch ....................................................................................... 5-58
TDSCDMA_RefChDecoder ................................................................................... 5-62
TDSCDMA_RM_Cal ............................................................................................. 5-72
TDSCDMA_SubFrameSeg ................................................................................... 5-74
TDSCDMA_TFCI_Encoder ................................................................................... 5-76
TDSCDMA_TrChDeMux ....................................................................................... 5-78
TDSCDMA_TrChMux .......................................................................................... 5-83

6 Physical Channel Components

TDSCDMA_DPCH ................................................................................................. 6-2
TDSCDMA_DwPCH ............................................................................................. 6-9
TDSCDMA_FPACH ............................................................................................. 6-12
TDSCDMA_PCCPCH .......................................................................................... 6-14
TDSCDMA_PICH ................................................................................................. 6-16
TDSCDMA_PRACH ............................................................................................. 6-20
TDSCDMA_PSCCH ............................................................................................. 6-22
TDSCDMA_SCCPCH .......................................................................................... 6-25
TDSCDMA_UpPCH ............................................................................................. 6-27

7 Receivers
8 Signal Sources

TDSCDMA_DL_RF .......................................................... 8-2
TDSCDMA_DL_Src ...................................................... 8-9
TDSCDMA_OCNS ....................................................... 8-11
TDSCDMA_SlotSrc .................................................... 8-13
TDSCDMA_UL_RF ...................................................... 8-17
TDSCDMA_UL_Src ..................................................... 8-23

Index
Chapter 1: TDSCDMA Design Library

Introduction

TD-SCDMA is a Chinese contribution to the international family of Mobile Radio Systems for 3G services of UMTS and IMT 2000. It is now one option of UTRA-TDD, called 1.28Mcps TDD or low chip rate (LCR) TDD. It is an advanced CDMA/TDMA/TDD system with an adaptive synchronous operation.

TD-SCDMA system simulation models based on the 3GPP TDD LCR standard demonstrate signal generation and receiving capabilities; basic measurements are considered. TD-SCDMA aligns with the same version of the specification used by the Agilent ESG-C, PSA II and VSA.

Physical Layer

The frame structure, illustrated in Figure 1-1, recognizes new smart antenna and uplink synchronization technologies.

![Figure 1-1. Physical Channel Signal Format](image-url)
Uplink and downlink time slots in each frame are separated by a switching point. There are two switching points in each sub-frame: TS0 is always allocated as downlink; TS1 is always allocated as uplink. There are three special time slots:

- DwPTS: downlink pilot time slot, 96 chip duration.
- UpPTS: uplink pilot time slot, 160 chip duration.
- GP: main guard period for TDD operation, 96 chip duration.

The system can operate on symmetric and asymmetric modes by properly configuring the number of downlink and uplink time slots.

The burst structure is illustrated in Figure 1-2.

The transmitter structure of a physical channel is illustrated in Figure 1-3.
Physical channels have a 3-layer structure.

- Time slot: 675 µsec slot consisting of a number of Symbols. Time slots are used in a TDMA component to separate different user signals in time and code domain.
- Radio frame: 5 msec frame consisting of 7 time slots.
- System frame numbering.

**Component Libraries**

The TD-SCDMA Design Library consists of behavioral models and subnetworks organized in libraries that are described in the following sections.

**Fully-Coded Source Components**

Fully-coded source library components provide fully-coded downlink and uplink sources according to the reference measurement channel specifications.
TDSCDMA Design Library

**Measurements Components**

Measurements library components measure BER/BLER, EVM, constellation, complementary cumulative distribution function and RF power, and provide multipath fading channels.

- TDSCDMA_BER calculates the BER and BLER by comparing the two input signals.
- TDSCDMA_Constellation measures the constellation of the received signal.
- TDSCDMA_EVM measures the EVM of the input signal.
- TDSCDMA_RF_CCDF measures the CCDF of the RF signal.
- TDSCDMA_RF_PwrMeasure measures RF signal average power and power vs. time.
- TDSCDMA_FWDChannel and TDSCDMA_REVChannel are the multipath fading channels for forward and reverse links, respectively. The profile of the channel is according to 3GPP TDD specifications.

**Modems Components**

Modems library components provide modulation, OVSF and spreading code generation, synchronization and midamble code generation, burst and frame generation.

- TDSCDMA_BurstMux generates a burst in a physical channel.
- TDSCDMA_DPCH_DataMux multiplexes TFCI, SS, and TPC data in a dedicated physical channel.
- TDSCDMA_Midamble generates midamble codes.
- TDSCDMA_Modulator performs QPSK and 8PSK modulation.
- TDSCDMA_OnePhyCh generates a sub-frame in a physical channel.
- TDSCDMA_OVSF generates OVSF codes.
- TDSCDMA_PSCH_DataMux multiplexes time division data in a shared physical channel.
- TDSCDMA_Scramble generates scramble codes.
- TDSCDMA_Sync generates synchronization codes.
Multiplexing and Coding Components

The Multiplexing and Coding library components include interleaving, rate matching, channel coding, and physical channel mapping.

- TDSCDMA_TFCl_Encoder encodes TFCI bits into TFCI code words.
- TDSCDMA_1stIntlvr, TDSCDMA_1stDeIntlvr, TDSCDMA_2ndIntlvr and TDSCDMA_2ndDeIntlvr are the first and the second interleavers and de-interleavers, respectively.
- TDSCDMA_CRC_Encoder and TDSCDMA_CRC_Decoder are the CRC encoder and the decoder, respectively.
- TDSCDMA_ChCoding and TDSCDMA_ChDecoding are the channel encoder and the decoder, respectively; coding schemes can be convolutional and Turbo.
- TDSCDMA_RateMatch and TDSCDMA_DeRateMatch provide rate match and de-match, respectively, for physical channels.
- TDSCDMA_RefChDecoder is a sub-network which implements a complete decoding process after demodulation for reference measurement channels.

Physical Channels Components

Physical Channels library Components generate physical channel signals.

- TDSCDMA_DPCH generates dedicated physical channel signals.
- TDSCDMA_DwPCH generates downlink synchronization channel signals.
- TDSCDMA_FPACH generates fast physical access channel signals.
- TDSCDMA_PCCPCH generates primary common control physical channel signals.
- TDSCDMA_PICH generates page indicator channel signals.
- TDSCDMA_PRACH generates physical random access channel signals.
- TDSCDMA_PSCCH generates physical downlink/uplink shared channel signals.
- TDSCDMA_SCCPCH generates secondary common control physical channel signals.
- TDSCDMA_UpPCH generates uplink synchronization channel signals.
Receivers

Receiver library components configure Rake and joint detection (JD) receivers.

- TDSCDMA_ChannelEstimation implements channel estimation for both Rake and joint detection receivers.
- TDSCDMA_12_2_DL_JD_Receiver is a joint detection receiver for 12.2 kbps downlink reference channel with 8 DPCH0.
- TDSCDMA_12_2_UL_JD_Receiver is a joint detection receiver for 12.2 kbps uplink reference channel with 4 DPCH0.
- TDSCDMA_12_2_DL_RakeReceiver and TDSCDMA_12_2_UL_RakeReceiver are Rake receivers for 12.2 kbps downlink and uplink reference channels, respectively.

Signal Sources

Signal Sources library components generate uplink and downlink signal sources.

- TDSCDMA_DL_RF generates downlink RF signals of DPCH with a 12.2 kbps data rate.
- TDSCDMA_DL_Src generates downlink baseband signals of DPCH with a 12.2 kbps data rate.
- TDSCDMA_UL_RF generates uplink RF signals of DPCH with a 12.2 kbps data rate.
- TDSCDMA_UL_Src generates uplink baseband signals of DPCH with a 12.2 kbps data rate.

Glossary of Terms

<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>TD-SCDMA</td>
<td>time division - synchronization code division multi-access</td>
</tr>
<tr>
<td>8PSK</td>
<td>8-ary phase shift keying</td>
</tr>
<tr>
<td>ACLR</td>
<td>adjacent channel leakage ratio</td>
</tr>
<tr>
<td>BER</td>
<td>bit error ratio</td>
</tr>
<tr>
<td>BLER</td>
<td>block error ratio</td>
</tr>
<tr>
<td>bps</td>
<td>bits per second</td>
</tr>
</tbody>
</table>

Table 1-1. Glossary of Terms
<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>CCDF</td>
<td>complementary cumulative distribution function</td>
</tr>
<tr>
<td>CDMA</td>
<td>code division multiple access</td>
</tr>
<tr>
<td>DpCH</td>
<td>dedicated physical channel</td>
</tr>
<tr>
<td>DwPCH</td>
<td>downlink pilot channel</td>
</tr>
<tr>
<td>DwPTS</td>
<td>downlink pilot time slot</td>
</tr>
<tr>
<td>FPACH</td>
<td>fast physical access channel</td>
</tr>
<tr>
<td>GP</td>
<td>guard period</td>
</tr>
<tr>
<td>Jd</td>
<td>joint detection</td>
</tr>
<tr>
<td>Lcr</td>
<td>low chip rate</td>
</tr>
<tr>
<td>OVSF</td>
<td>orthogonal variable spreading factor</td>
</tr>
<tr>
<td>PCCPCH</td>
<td>primary common control physical channel</td>
</tr>
<tr>
<td>PDSCCH</td>
<td>physical downlink shared channel</td>
</tr>
<tr>
<td>PICH</td>
<td>page indicator channel</td>
</tr>
<tr>
<td>PRACH</td>
<td>physical random access channel</td>
</tr>
<tr>
<td>PUSCH</td>
<td>physical uplink shared channel</td>
</tr>
<tr>
<td>QPSK</td>
<td>quadrature phase shift keying</td>
</tr>
<tr>
<td>SCCPCH</td>
<td>secondary common control physical channel</td>
</tr>
<tr>
<td>TDD</td>
<td>time division duplex</td>
</tr>
<tr>
<td>TFCI</td>
<td>transmit format combination indicator</td>
</tr>
<tr>
<td>UpPCH</td>
<td>uplink pilot channel</td>
</tr>
<tr>
<td>UpPTS</td>
<td>uplink pilot time slot</td>
</tr>
</tbody>
</table>
Chapter 2: Fully Coded Sources
Fully Coded Sources

**TDSCDMA_RefCh**

**Description**
Reference measurement channel

**Library**
TDSCDMA, Fully Coded Source

**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>Link</td>
<td>link selection: Downlink, Uplink</td>
<td>Downlink</td>
<td>enum</td>
<td></td>
</tr>
<tr>
<td>RefCh</td>
<td>reference channel selection indicator: CH_12.2k_MultiCode, CH_12.2k_SingleCode, CH_64k, CH_144k, CH_384k</td>
<td>CH_12.2k_MultiCode</td>
<td>enum</td>
<td></td>
</tr>
<tr>
<td>PhyChNum_SA</td>
<td>physical channel allocation configuration</td>
<td>0 0 2 0 0 0 0</td>
<td>int array</td>
<td>[0, 16] for Downlink, [0, 2] for Uplink</td>
</tr>
<tr>
<td>MaxPhyChNum</td>
<td>sum of allocated physical channel in all slots</td>
<td>2</td>
<td>int</td>
<td>[1, 112]</td>
</tr>
<tr>
<td>ModType_SA</td>
<td>type of modulation of all slots, 0 for QPSK, 1 for 8PSK</td>
<td>0 0 0 0 0 0</td>
<td>int array</td>
<td>[0, 1]</td>
</tr>
<tr>
<td>TFCI_SA</td>
<td>allocated TFCI transmitted active slots configuration</td>
<td>0 0 1 0 0 0 0</td>
<td>int array</td>
<td>[0, 1]</td>
</tr>
<tr>
<td>TFCI_Length_SA</td>
<td>length of TFCI of all slots</td>
<td>0 0 16 0 0 0 0</td>
<td>int array</td>
<td>[0, 4, 8, 16, 32] for QPSK, [0, 6, 12, 24, 48] for 8PSK</td>
</tr>
<tr>
<td>SS_TPC_SA</td>
<td>type of SS and TPC of all slots</td>
<td>2 2 1 2 2 2 2</td>
<td>int array</td>
<td>[1, 2, 3]</td>
</tr>
<tr>
<td>MinSF_PA</td>
<td>minimum spreading factor array corresponding to allocated physical channels</td>
<td>16 16</td>
<td>int array</td>
<td>[1, 16] for Downlink, [1, 2, 4, 8, 16] for Uplink</td>
</tr>
<tr>
<td>SpreadCode_PA</td>
<td>index of OVSF code corresponding to allocated physical channels</td>
<td>1 2</td>
<td>int array</td>
<td></td>
</tr>
</tbody>
</table>
BasicMidambleID | index of basic midamble | 1 | int
K_SA | maximum number of midamble shifts in a cell for all slots | 16 16 16 16 16 16 16 | int array
MidambleID_SA | index of midamble for all slots | 5 5 5 5 5 5 | int array
Gain_PA | gain setting array corresponding to allocated physical channels | 1.0 1.0 | real array

**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>DCH</td>
<td>DCH data out</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>OutI</td>
<td>out</td>
<td>real</td>
</tr>
<tr>
<td>3</td>
<td>OutQ</td>
<td>out</td>
<td>real</td>
</tr>
</tbody>
</table>

**Notes/Equations**

1. This subnetwork implements a reference measurement channel.

   The schematic for this subnetwork is shown in Figure 2-1.
2. MinSF_PA indicates the minimum spread factor that can be used for corresponding physical channel. The size of MinSF_PA must be equal to the sum of PhyChNum_SA elements.

3. SS_TPC_SA indicates the number of SS and TPC symbols transmitted in each slot: 1 denotes one SS and one TPC symbols are transmitted; 2 denotes no SS and no TPC symbols are transmitted; 3 denotes 16/SF SS and 16/SF TPC symbols are transmitted, where SF is the spreading factor of the physical channel used to transmitted SS and TPC symbols. Only the first allocated physical channel in each slot is used to transmit SS and TPC symbols.

4. The structure and settings for the various data rates are given in Table 2-1 through Table 2-8.

### Table 2-1. 12.2 kbps UL Reference Measurement Channel

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Information data rate</td>
<td>12.2 kbps</td>
</tr>
<tr>
<td>RUss allocated</td>
<td>1TS (1 x SF8) = 2RU/5ms</td>
</tr>
<tr>
<td>Midamble</td>
<td>144</td>
</tr>
<tr>
<td>Interleaving</td>
<td>20 ms</td>
</tr>
</tbody>
</table>

2-4 TDSCDMA_RefCh
Table 2-1. 12.2 kbps UL Reference Measurement Channel

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Power control</td>
<td>4 Bit/user/10ms</td>
</tr>
<tr>
<td>TFCI</td>
<td>16 Bit/user/10ms</td>
</tr>
<tr>
<td>4 Bit reserved for future use (place of SS)</td>
<td>4 Bit/user/10ms</td>
</tr>
<tr>
<td>Inband signalling DCCH</td>
<td>2.4 kbps</td>
</tr>
<tr>
<td>Puncturing level at code rate 1/3: DCH / DCCH</td>
<td>33% / 33%</td>
</tr>
</tbody>
</table>

Table 2-2. 12.2 kbps UL and DL Multi-Code Reference Measurement Channel

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Information data rate</td>
<td>12.2 kbps</td>
</tr>
<tr>
<td>RU’s allocated</td>
<td>1TS (2 × SF16) – 2RU/5ms</td>
</tr>
<tr>
<td>Midamble</td>
<td>144</td>
</tr>
<tr>
<td>Interleaving</td>
<td>20 ms</td>
</tr>
<tr>
<td>Power control (TPC)</td>
<td>4 Bit/user/10ms</td>
</tr>
<tr>
<td>TFCI</td>
<td>16 Bit/user/10ms</td>
</tr>
<tr>
<td>Synchronization shift (SS)</td>
<td>4 Bit/user/10ms</td>
</tr>
<tr>
<td>Inband signalling DCCH</td>
<td>2.4 kbps</td>
</tr>
<tr>
<td>Puncturing level at code rate 1/3: DCH / DCCH</td>
<td>33% / 33%</td>
</tr>
</tbody>
</table>
Fully Coded Sources

Table 2-3. 64 kbps UL Reference Measurement Channel

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Information data rate</td>
<td>64 kbps</td>
</tr>
<tr>
<td>RUs allocated</td>
<td>1TS (1 × SF2) = 8RU/5ms</td>
</tr>
<tr>
<td>Midamble</td>
<td>144</td>
</tr>
<tr>
<td>Interleaving</td>
<td>20 ms</td>
</tr>
<tr>
<td>Power control (TPC)</td>
<td>4 Bit/user/10ms</td>
</tr>
<tr>
<td>TFCI</td>
<td>16 Bit/user/10ms</td>
</tr>
<tr>
<td>Synchronization shift (SS)</td>
<td>4 Bit/user/10ms</td>
</tr>
<tr>
<td>Inband signalling DCCH</td>
<td>2.4 kbps</td>
</tr>
<tr>
<td>Puncturing level at code rate: 1/3 DCH / DCCH</td>
<td>32% / 0</td>
</tr>
</tbody>
</table>

Table 2-4. 64 kbps DL Reference Measurement Channel

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Information data rate</td>
<td>64 kbps</td>
</tr>
<tr>
<td>RUs allocated</td>
<td>1TS (8 × SF16) = 8RU/5ms</td>
</tr>
<tr>
<td>Midamble</td>
<td>144</td>
</tr>
<tr>
<td>Interleaving</td>
<td>20 ms</td>
</tr>
<tr>
<td>Power control (TPC)</td>
<td>4 Bit/user/10ms</td>
</tr>
<tr>
<td>TFCI</td>
<td>16 Bit/user/10ms</td>
</tr>
<tr>
<td>Synchronization shift (SS)</td>
<td>4 Bit/user/10ms</td>
</tr>
<tr>
<td>Inband signalling DCCH</td>
<td>2.4 kbps</td>
</tr>
<tr>
<td>Puncturing level at code rate: 1/3 DCH / DCCH</td>
<td>32% / 0</td>
</tr>
</tbody>
</table>
Table 2-5. 144 kbps UL Reference Measurement Channel

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Information data rate</td>
<td>144 kbps</td>
</tr>
<tr>
<td>RUs allocated</td>
<td>$2TS \times (1 \times SF2) = 16RU/5ms$</td>
</tr>
<tr>
<td>Midamble</td>
<td>144</td>
</tr>
<tr>
<td>Interleaving</td>
<td>20 ms</td>
</tr>
<tr>
<td>Power control (TPC)</td>
<td>8 Bit/user/10ms</td>
</tr>
<tr>
<td>TFCI</td>
<td>32 Bit/user/10ms</td>
</tr>
<tr>
<td>Synchronization shift (SS)</td>
<td>8 Bit/user/10ms</td>
</tr>
<tr>
<td>Inband signalling DCCH</td>
<td>2.4 kbps</td>
</tr>
<tr>
<td>Puncturing level at code rate: 1/3 DCH / DCCH</td>
<td>38% / 7%</td>
</tr>
</tbody>
</table>

Table 2-6. 144 kbps DL Reference Measurement Channel

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Information data rate</td>
<td>144 kbps</td>
</tr>
<tr>
<td>RUs allocated</td>
<td>$2TS \times (8 \times SF16) = 16RU/5ms$</td>
</tr>
<tr>
<td>Midamble</td>
<td>144</td>
</tr>
<tr>
<td>Interleaving</td>
<td>20 ms</td>
</tr>
<tr>
<td>Power control (TPC)</td>
<td>8 Bit/user/10ms</td>
</tr>
<tr>
<td>TFCI</td>
<td>32 Bit/user/10ms</td>
</tr>
<tr>
<td>Synchronization shift (SS)</td>
<td>8 Bit/user/10ms</td>
</tr>
<tr>
<td>Inband signalling DCCH</td>
<td>2.4 kbps</td>
</tr>
<tr>
<td>Puncturing level at code rate: 1/3 DCH / DCCH</td>
<td>38% / 7%</td>
</tr>
</tbody>
</table>
5. The configuration for transport channels is fixed when the Link and RefCh parameter are set. The configuration for physical channels can be set flexibly according to Table 2-1 through Table 2-8. An example for each configuration is shown in Table 2-9 through Table 2-16.
### Table 2-9. 12.2 kbps UL Reference Measurement
Physical Channel Setting

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Link</td>
<td>Uplink</td>
</tr>
<tr>
<td>RefCh</td>
<td>12.2K_SingleCode</td>
</tr>
<tr>
<td>PhyChNum_SA</td>
<td>0 0 1 0 0 0 0</td>
</tr>
<tr>
<td>MaxPhyChNum</td>
<td>1</td>
</tr>
<tr>
<td>ModType_SA</td>
<td>0[7]</td>
</tr>
<tr>
<td>TFCI_SA</td>
<td>0 0 1 0 0 0 0</td>
</tr>
<tr>
<td>TFCI_Length_SA</td>
<td>0 0 1 6 0 0 0 0</td>
</tr>
<tr>
<td>SS_TPC_SA</td>
<td>2 2 1 2 2 2 2 2</td>
</tr>
<tr>
<td>MinSF_PA</td>
<td>8</td>
</tr>
<tr>
<td>SpreadCode_PA</td>
<td>1</td>
</tr>
<tr>
<td>Gain_PA</td>
<td>1.0</td>
</tr>
</tbody>
</table>

### Table 2-10. 12.2 kbps UL and DL Multi-Code Reference Measurement
Physical Channel Setting

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Link</td>
<td>Uplink for UL and Downlink for DL</td>
</tr>
<tr>
<td>RefCh</td>
<td>12.2K_MultiCode</td>
</tr>
<tr>
<td>PhyChNum_SA</td>
<td>0 0 2 0 0 0 0 0</td>
</tr>
<tr>
<td>MaxPhyChNum</td>
<td>2</td>
</tr>
<tr>
<td>ModType_SA</td>
<td>0[7]</td>
</tr>
<tr>
<td>TFCI_SA</td>
<td>0 0 1 0 0 0 0</td>
</tr>
<tr>
<td>TFCI_Length_SA</td>
<td>0 0 1 6 0 0 0 0</td>
</tr>
<tr>
<td>SS_TPC_SA</td>
<td>2 2 1 2 2 2 2 2</td>
</tr>
<tr>
<td>MinSF_PA</td>
<td>16[2]</td>
</tr>
<tr>
<td>SpreadCode_PA</td>
<td>1 3</td>
</tr>
<tr>
<td>Gain_PA</td>
<td>1.0[2]</td>
</tr>
</tbody>
</table>

### Table 2-11. 64 kbps UL Reference Measurement
Physical Channel Setting

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Link</td>
<td>Uplink</td>
</tr>
<tr>
<td>RefCh</td>
<td>64K</td>
</tr>
</tbody>
</table>
Fully Coded Sources

Table 2-11. 64 kbps UL Reference Measurement Physical Channel Setting (continued)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>PhyChNum_SA</td>
<td>0 0 1 0 0 0 0</td>
</tr>
<tr>
<td>MaxPhyChNum</td>
<td>1</td>
</tr>
<tr>
<td>ModType_SA</td>
<td>0[7]</td>
</tr>
<tr>
<td>TFCI_SA</td>
<td>0 0 1 0 0 0 0</td>
</tr>
<tr>
<td>TFCI_Length_SA</td>
<td>0 0 1 6 0 0 0</td>
</tr>
<tr>
<td>SS_TPC_SA</td>
<td>2 2 1 2 2 2 2</td>
</tr>
<tr>
<td>MinSF_PA</td>
<td>2</td>
</tr>
<tr>
<td>SpreadCode_PA</td>
<td>1</td>
</tr>
<tr>
<td>Gain_PA</td>
<td>1.0</td>
</tr>
</tbody>
</table>

Table 2-12. 64 kbps DL Reference Measurement Physical Channel Setting

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Link</td>
<td>Downlink</td>
</tr>
<tr>
<td>RefCh</td>
<td>64K</td>
</tr>
<tr>
<td>PhyChNum_SA</td>
<td>0 0 8 0 0 0 0</td>
</tr>
<tr>
<td>MaxPhyChNum</td>
<td>8</td>
</tr>
<tr>
<td>ModType_SA</td>
<td>0[7]</td>
</tr>
<tr>
<td>TFCI_SA</td>
<td>0 0 1 0 0 0 0</td>
</tr>
<tr>
<td>TFCI_Length_SA</td>
<td>0 0 1 6 0 0 0</td>
</tr>
<tr>
<td>SS_TPC_SA</td>
<td>2 2 1 2 2 2 2</td>
</tr>
<tr>
<td>MinSF_PA</td>
<td>16[8]</td>
</tr>
<tr>
<td>SpreadCode_PA</td>
<td>1 3 5 7 9 1 1 1 3 1 5</td>
</tr>
<tr>
<td>Gain_PA</td>
<td>1.0[8]</td>
</tr>
</tbody>
</table>

Table 2-13. 144 kbps UL Reference Measurement Channel Physical Channel Setting

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Link</td>
<td>Uplink</td>
</tr>
<tr>
<td>RefCh</td>
<td>144K</td>
</tr>
<tr>
<td>PhyChNum_SA</td>
<td>0 0 1 1 0 0 0</td>
</tr>
<tr>
<td>MaxPhyChNum</td>
<td>2</td>
</tr>
</tbody>
</table>
Table 2-13. 144 kbps UL Reference Measurement Channel Physical Channel Setting (continued)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>ModType_SA</td>
<td>0[7]</td>
</tr>
<tr>
<td>TFCI_SA</td>
<td>0 0 1 1 0 0 0</td>
</tr>
<tr>
<td>TFCI_Length_SA</td>
<td>0 0 16 16 0 0 0</td>
</tr>
<tr>
<td>SS_TPC_SA</td>
<td>2 2 1 1 2 2 2</td>
</tr>
<tr>
<td>MinSF_PA</td>
<td>2[2]</td>
</tr>
<tr>
<td>SpreadCode_PA</td>
<td>1 2</td>
</tr>
<tr>
<td>Gain_PA</td>
<td>1.0[2]</td>
</tr>
</tbody>
</table>

Table 2-14. 144 kbps DL Reference Measurement Physical Channel Setting

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Link</td>
<td>Downlink</td>
</tr>
<tr>
<td>RefCh</td>
<td>144K</td>
</tr>
<tr>
<td>PhyChNum_SA</td>
<td>0 0 8 8 0 0</td>
</tr>
<tr>
<td>MaxPhyChNum</td>
<td>16</td>
</tr>
<tr>
<td>ModType_SA</td>
<td>0[7]</td>
</tr>
<tr>
<td>TFCI_SA</td>
<td>0 0 1 1 0 0 0</td>
</tr>
<tr>
<td>TFCI_Length_SA</td>
<td>0 0 16 16 0 0 0</td>
</tr>
<tr>
<td>SS_TPC_SA</td>
<td>2 2 1 1 2 2 2</td>
</tr>
<tr>
<td>MinSF_PA</td>
<td>16[16]</td>
</tr>
<tr>
<td>SpreadCode_PA</td>
<td>1 5 7 9 1 1 3 1 5 2 4 6 8 10 12 14 16</td>
</tr>
<tr>
<td>Gain_PA</td>
<td>1.0[16]</td>
</tr>
</tbody>
</table>

Table 2-15. 384 kbps UL Reference Measurement Physical Channel Setting

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Link</td>
<td>Uplink</td>
</tr>
<tr>
<td>RefCh</td>
<td>384K</td>
</tr>
<tr>
<td>PhyChNum_SA</td>
<td>0 0 2 2 2 2 0</td>
</tr>
<tr>
<td>MaxPhyChNum</td>
<td>8</td>
</tr>
<tr>
<td>ModType_SA</td>
<td>0[7]</td>
</tr>
<tr>
<td>TFCI_SA</td>
<td>0 0 1 1 1 1 0</td>
</tr>
</tbody>
</table>
Table 2-15. 384 kbps DL Reference Measurement Physical Channel Setting (continued)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>TFCl_Length_SA</td>
<td>0 0 16 16 16 0</td>
</tr>
<tr>
<td>SS_TPC_SA</td>
<td>2 2 1 1 1 2</td>
</tr>
<tr>
<td>MinSF_PA</td>
<td>8 2 8 2 8 2</td>
</tr>
<tr>
<td>SpreadCode_PA</td>
<td>1 2 1 2 1 2</td>
</tr>
<tr>
<td>Gain_PA</td>
<td>1.0[8]</td>
</tr>
</tbody>
</table>

Table 2-16. 384 kbps UL Reference Measurement Physical Channel Setting (continued)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Link</td>
<td>Downlink</td>
</tr>
<tr>
<td>RefCh</td>
<td>384K</td>
</tr>
<tr>
<td>PhyChNum_SA</td>
<td>0 0 10 10 10 0</td>
</tr>
<tr>
<td>MaxPhyChNum</td>
<td>1</td>
</tr>
<tr>
<td>ModType_SA</td>
<td>0[7]</td>
</tr>
<tr>
<td>TFCl_SA</td>
<td>0 0 1 1 1 1 0</td>
</tr>
<tr>
<td>TFCl_Length_SA</td>
<td>0 0 16 16 16 16 0</td>
</tr>
<tr>
<td>SS_TPC_SA</td>
<td>2 2 1 1 1 2</td>
</tr>
<tr>
<td>MinSF_PA</td>
<td>16[40]</td>
</tr>
<tr>
<td>SpreadCode_PA</td>
<td>1 3 5 7 9 11 13 15 2 4 6 8 10 12 14 16 1 3 5 7 9 11 13 15 2 4 6 8 10 12 14 16 1 3 5 7 9 11 13 15</td>
</tr>
<tr>
<td>Gain_PA</td>
<td>1.0[40]</td>
</tr>
</tbody>
</table>

References


TDSCDMA_RefCh_RF

Description  RF reference measurement channel
Library  TDSCDMA, Fully Coded Source
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>ROut</td>
<td>output resistance</td>
<td>DefaultROut</td>
<td>Ohm</td>
<td>real</td>
<td>(0, ∞)</td>
</tr>
<tr>
<td>FCarrier</td>
<td>carrier frequency</td>
<td>1900MHz</td>
<td>Hz</td>
<td>real</td>
<td>(0, ∞)</td>
</tr>
<tr>
<td>VRef</td>
<td>reference voltage</td>
<td>0.5222V</td>
<td>V</td>
<td>real</td>
<td>(0, ∞)</td>
</tr>
<tr>
<td>Power</td>
<td>modulator output power</td>
<td>0.1W</td>
<td>W</td>
<td>real</td>
<td></td>
</tr>
<tr>
<td>FilterLength</td>
<td>length of raised cosine filters in number of symbols</td>
<td>16</td>
<td>int</td>
<td></td>
<td>(0, ∞)</td>
</tr>
<tr>
<td>SamplesPerSymbol</td>
<td>samples per symbol period</td>
<td>8</td>
<td>int</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Link</td>
<td>link selection: Downlink, Uplink</td>
<td>Downlink</td>
<td>enum</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RefCh</td>
<td>reference channel selection indicator: CH_12.2k_MultiCode, CH_12.2k_SingleCode, CH_64k, CH_144k, CH_384k</td>
<td>CH_12.2k_MultiCode</td>
<td>enum</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PhyChNum_SA</td>
<td>physical channel allocation configuration</td>
<td>0 0 2 0 0 0 0</td>
<td>int array</td>
<td>[0, 16] for Downlink, [0, 2] for Uplink</td>
<td></td>
</tr>
<tr>
<td>MaxPhyChNum</td>
<td>sum of allocated physical channel in all slots</td>
<td>2</td>
<td>int</td>
<td></td>
<td>[1, 112]</td>
</tr>
<tr>
<td>ModType_SA</td>
<td>type of modulation of all slots, 0 for QPSK, 1 for 8PSK</td>
<td>0 0 0 0 0 0 0</td>
<td>int array</td>
<td>[0, 1]</td>
<td></td>
</tr>
<tr>
<td>TFCI_SA</td>
<td>allocated TFCI transmitted active slots configuration</td>
<td>0 0 1 0 0 0 0</td>
<td>int array</td>
<td>[0, 1]</td>
<td></td>
</tr>
</tbody>
</table>
Fully Coded Sources

<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>TFCI_Length_SA</td>
<td>length of TFCI of all slots</td>
<td>0 0 16 0 0 0 0</td>
<td>int array</td>
<td>[0, 4, 8, 16, 32] for QPSK, [0, 6, 12, 24, 48] for 8PSK</td>
<td></td>
</tr>
<tr>
<td>SS_TPC_SA</td>
<td>type of SS and TPC of all slots</td>
<td>2 2 1 2 2 2 2</td>
<td>int array</td>
<td>[1, 2, 3]</td>
<td></td>
</tr>
<tr>
<td>MinSF_PA</td>
<td>minimum spreading factor array corresponding to allocated physical channels</td>
<td>16 16</td>
<td>int array</td>
<td>(1, 16) for Downlink [1, 2, 4, 8, 16] for Uplink</td>
<td></td>
</tr>
<tr>
<td>SpreadCode_PA</td>
<td>index of OVSF code corresponding to allocated physical channels</td>
<td>1 2</td>
<td>int array</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BasicMidambleID</td>
<td>index of basic midamble</td>
<td>0</td>
<td>int</td>
<td></td>
<td></td>
</tr>
<tr>
<td>K_SA</td>
<td>maximum number of midamble shifts in a cell for all slots</td>
<td>16 16 16 16 16 16</td>
<td>int array</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MidambleID_SA</td>
<td>index of midamble for all slots</td>
<td>5 5 5 5 5 5 5 5</td>
<td>int array</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Gain_PA</td>
<td>gain setting array corresponding to allocated physical channels</td>
<td>1.0 1.0</td>
<td>real array</td>
<td></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>sig</td>
<td>output signal</td>
<td>timed</td>
</tr>
<tr>
<td>2</td>
<td>bits</td>
<td>information bits</td>
<td>int</td>
</tr>
</tbody>
</table>

Notes/Equations

1. This subnetwork implements RF reference measurement channel.

   The schematic for this subnetwork is shown in Figure 2-2.
2. This subnetwork supports both uplink and downlink channels with data rates from 12.2k to 384k. VRef must be set according to data rate in order to obtain the desired output power. Parameter settings for reference channel can be referred to the TDSCDMA_RefCh.
Fully Coded Sources
Chapter 3: Measurements
Measurements

TDSCDMA_BER

Description  BER and BLER measurement
Library       TDSCDMA, Measurements
Class         SDFTDSCDMA_BER
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>BlockLength</td>
<td>block length</td>
<td>244</td>
<td>int</td>
<td>[1, 5000]</td>
</tr>
<tr>
<td>IgnoreNumber</td>
<td>number of initially ignored firings</td>
<td>0</td>
<td>int</td>
<td>[0, 1000]</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>input1</td>
<td>input data 1</td>
<td>int</td>
</tr>
<tr>
<td>2</td>
<td>input2</td>
<td>input data 2</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>
<tr>
<td>4</td>
<td>BLER</td>
<td>block error rate</td>
<td>real</td>
</tr>
<tr>
<td>5</td>
<td>BlkNum</td>
<td>number of blocks</td>
<td>int</td>
</tr>
</tbody>
</table>

Notes/Equations

1. This model is used to measure BER and BLER.
   Each firing, 1 BER token, 1 BLER token and 1 Block token are produced when BlockLength Input1 and Input2 Output tokens are consumed.
TDSCDMA_Constellation

Description  Constellation of received data
Library   TDSCDMA, Measurements
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>RLoad</td>
<td>reference resistance</td>
<td>DefaultRln</td>
<td>Ohm</td>
<td>real</td>
<td>(0, ∞          )</td>
</tr>
<tr>
<td>RTemp</td>
<td>temperature of reference resistor, in degrees C</td>
<td>DefaultRTemp</td>
<td></td>
<td>real</td>
<td>(-273.15, ∞)</td>
</tr>
<tr>
<td>FCarrier</td>
<td>carrier frequency</td>
<td>1900MHz</td>
<td>Hz</td>
<td>real</td>
<td>(-1) or (0, ∞)†</td>
</tr>
<tr>
<td>AnalysisTimeslot</td>
<td>timeslot to be analyzed: T0, T1, T2, T3, T4, T5, T6</td>
<td>TS2</td>
<td></td>
<td>enum</td>
<td></td>
</tr>
<tr>
<td>SamplesPerSymbol</td>
<td>samples per symbol</td>
<td>8</td>
<td>int</td>
<td>[1, 32]</td>
<td></td>
</tr>
<tr>
<td>FilterLength</td>
<td>length of raised cosine filters in number of symbols</td>
<td>16</td>
<td>int</td>
<td>(0, ∞)</td>
<td></td>
</tr>
<tr>
<td>SubframesToMeasure</td>
<td>number of subframes to be measured</td>
<td>1</td>
<td>int</td>
<td>[1, 65535]</td>
<td></td>
</tr>
<tr>
<td>SyncCodeUsed</td>
<td>Code used in synchronization: DwPTS, UpPTS, Midamble</td>
<td>DwPTS</td>
<td></td>
<td>enum</td>
<td></td>
</tr>
<tr>
<td>SyncCodeIdx</td>
<td>index of basic synchronization code</td>
<td>0</td>
<td>int</td>
<td>[0, 31] when SyncCodeUsed = DwPTS; [0, 255] when SyncCodeUsed = UpPTS</td>
<td></td>
</tr>
<tr>
<td>ModPhase</td>
<td>type of modulation quadruples, valid only in downlink: S1, S2</td>
<td>S1</td>
<td></td>
<td>enum</td>
<td></td>
</tr>
<tr>
<td>MidambleAllocScheme</td>
<td>midamble allocation scheme: UE_Specific, Common, Default</td>
<td>Default</td>
<td></td>
<td>enum</td>
<td></td>
</tr>
<tr>
<td>BasicMidambleID</td>
<td>index of basic midamble</td>
<td>0</td>
<td>int</td>
<td>[0, 127]</td>
<td></td>
</tr>
</tbody>
</table>

† This parameter has a range that extends to positive infinity, indicating it can be any positive value or infinity.
Measurements

1. This subnetwork is used to measure the TDSCDMA signal constellation. The schematic for this subnetwork is shown in Figure 3-1.

2. The TDSCDMA input signal delay is introduced by the filter (or device under test). The maximum delay that can be detected is the length of one subframe.

3. The received signal is synchronized and the information data is separated from the analyzed timeslot set by the AnalysisTimeslot parameter; the modulated symbols are despread from the information data and stored as a complex number. The constellation is determined by drawing the imaginary vs. the real part of the complex data stored.

4. AnalysisTimeslot specifies which timeslot is analyzed in the current measurement. It is also used to determine the frame boundary during synchronization when SyncCodeUsed is set to Midamble.

5. A raised-cosine filter is used in this subnetwork. FilterLength specifies the length of the filter; set this parameter to the same value as the signal source filter.

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 signals</td>
<td>timed</td>
</tr>
</tbody>
</table>

Notes/Equations

1. This subnetwork is used to measure the TDSCDMA signal constellation. The schematic for this subnetwork is shown in Figure 3-1.

2. The TDSCDMA input signal delay is introduced by the filter (or device under test). The maximum delay that can be detected is the length of one subframe.

3. The received signal is synchronized and the information data is separated from the analyzed timeslot set by the AnalysisTimeslot parameter; the modulated symbols are despread from the information data and stored as a complex number. The constellation is determined by drawing the imaginary vs. the real part of the complex data stored.

4. AnalysisTimeslot specifies which timeslot is analyzed in the current measurement. It is also used to determine the frame boundary during synchronization when SyncCodeUsed is set to Midamble.

5. A raised-cosine filter is used in this subnetwork. FilterLength specifies the length of the filter; set this parameter to the same value as the signal source filter.
6. SyncCodeUsed specifies the synchronization code.

- DwPTS (downlink pilot codes)  SyncCodeIdx and ModPhase synchronization-code-related parameters must be set.
- UpPTS (uplink pilot codes)  SyncCodeIdx synchronization-code-related parameter must be set.
- Midamble (midamble codes) MidambleAllocScheme, BasicMidambleD, K, MidambleD, PhyChNum, SpreadFactor and SpreadCode synchronization-code-related parameters must be set.

References

Measurements

**TDSCDMA_EVM**

**Description**  EVM measurement

**Library**  TDSCDMA, Measurements

**Class**  TSDF_TDSCDMA_EVM

**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>RLoad</td>
<td>load resistance. DefaultRLoad will inherit from the DF controller.</td>
<td>DefaultRLoad</td>
<td>Ohm</td>
<td>real</td>
<td>(0, ∞)</td>
</tr>
<tr>
<td>RTemp</td>
<td>physical temperature, in degrees C, of load resistance. DefaultRTemp will inherit from the DF controller.</td>
<td>DefaultRTemp</td>
<td>Celsius</td>
<td>real</td>
<td>[-273.15, ∞)</td>
</tr>
<tr>
<td>FCarrier</td>
<td>carrier frequency</td>
<td>1.9e9</td>
<td>Hz</td>
<td>real</td>
<td>(0, ∞)</td>
</tr>
<tr>
<td>Start</td>
<td>start time for data recording. DefaultTimeStart will inherit from the DF Controller.</td>
<td>DefaultTimeStart</td>
<td>sec</td>
<td>real</td>
<td>[0, ∞)</td>
</tr>
<tr>
<td>AverageType</td>
<td>average type: OFF, RMS (Video)</td>
<td>OFF</td>
<td>enum</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SubframesToAverage</td>
<td>number of subframes that will be averaged if AverageType is RMS (Video)</td>
<td>10</td>
<td>inf</td>
<td>[1, ∞)</td>
<td></td>
</tr>
<tr>
<td>ChipRate</td>
<td>chip rate</td>
<td>1.28e6</td>
<td>Hz</td>
<td>real</td>
<td>(0, ∞)</td>
</tr>
<tr>
<td>Alpha</td>
<td>root raised cosine filter roll off factor</td>
<td>0.22</td>
<td>real</td>
<td>[0.05, 1]</td>
<td></td>
</tr>
<tr>
<td>MirrorFrequencySpectrum</td>
<td>mirror frequency spectrum: NO, YES</td>
<td>NO</td>
<td>enum</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
This model performs an EVM measurement for a TD-SCDMA signal. The input signal must be a timed RF (complex envelope) signal or else the model will error out. The available results from this measurement are:

- **Avg_ChEVMrms_pct**: average channel EVM rms in %
- **ChEVMrms_pct**: channel EVM rms in % versus subframe
- **ChEVM_Pk_pct**: channel peak EVM in % versus subframe
- **ChEVM_Pk_symbol_idx**: channel peak EVM symbol index versus subframe

### 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>timed</td>
</tr>
</tbody>
</table>

### Notes/Equations

1. The model performs an EVM measurement for a TD-SCDMA signal. The input signal must be a timed RF (complex envelope) signal or else the model will error out. The available results from this measurement are:

   - **Avg_ChEVMrms_pct**: average channel EVM rms in %
   - **ChEVMrms_pct**: channel EVM rms in % versus subframe
   - **ChEVM_Pk_pct**: channel peak EVM in % versus subframe
   - **ChEVM_Pk_symbol_idx**: channel peak EVM symbol index versus subframe
Measurements

- **Avg_ChMagErr_rms_pct**: average channel magnitude error rms in %
- **ChMagErr_rms_pct**: channel magnitude error rms in % versus subframe
- **ChMagErr_Pk_pct**: channel peak magnitude error in % versus subframe
- **ChMagErr_Pk_symbol_idx**: channel peak magnitude error symbol index versus subframe
- **Avg_ChPhaseErr_deg**: average channel phase error in degrees
- **ChPhaseErr_deg**: channel phase error in degrees versus subframe
- **ChPhaseErr_Pk_deg**: channel peak phase error in degrees versus subframe
- **ChPhaseErr_Pk_symbol_idx**: channel peak phase error symbol index versus subframe
- **ChCodePhase_deg**: channel code phase (phase of the channel code with respect to the pilot) versus subframe
- **Avg_CompEVMrms_pct**: average composite EVM rms in %
- **CompEVMrms_pct**: composite EVM rms in % versus subframe
- **CompEVM_Pk_pct**: composite peak EVM in % versus subframe
- **CompEVM_Pk_chip_idx**: composite peak EVM chip index versus subframe
- **Avg_CompMagErr_rms_pct**: average composite magnitude error rms in %
- **CompMagErr_rms_pct**: composite magnitude error rms in % versus subframe
- **CompMagErr_Pk_pct**: composite peak magnitude error in % versus subframe
- **CompMagErr_Pk_chip_idx**: composite peak magnitude error chip index versus subframe
- **Avg_CompPhaseErr_deg**: average composite phase error in degrees
- **CompPhaseErr_deg**: composite phase error in degrees versus subframe
- **CompPhaseErr_Pk_deg**: composite peak phase error in degrees versus subframe
- **CompPhaseErr_Pk_chip_idx**: composite peak phase error chip index versus subframe
- **Avg_Rho**: average rho

---

3-8  TDSCDMA_EVM
• Rho: rho versus subframe
• Avg_FreqError_Hz: average frequency error in Hz
• FreqError_Hz: frequency error in Hz versus subframe
• Avg_IQ_Offset_dB: average IQ offset in dB
• IQ_Offset_dB: IQ offset in dB versus subframe
• Avg_QuadErr_deg: average quadrature error in degrees
• QuadErr_deg: quadrature error in degrees versus subframe
• Avg_GainImb_dB: average IQ gain imbalance in dB
• GainImb_dB: IQ gain imbalance in dB versus subframe

Results named with the Avg_prefix are results averaged over the number of subframes specified by the user (if AverageType is set to RMS (Video)). Results that are not named Avg_are results versus subframe. To use any of the results in an ael expression or in the Goal expression in an optimization setup, you must prefix them with the instance name of the model followed by a dot, for example T1.Avg_CompEVMrms_pct.

Following is a brief description of the algorithm used in this model and details of its parameter usage.

2. Starting at the time instant specified by the Start parameter, the model captures a signal segment of 10 msec and detects the beginning of a subframe (a 10 msec signal segment is guaranteed to contain a whole subframe). After the subframe is detected, the I and Q envelopes of the input signal are extracted. The FCarrier parameter sets the frequency of the internal local oscillator signal for the I and Q envelope extraction. Finally, the I and Q envelopes are passed to a complex algorithm that performs synchronization, demodulation, and EVM analysis. The algorithm that performs the synchronization, demodulation, and EVM analysis is the same as the one used in the Agilent 89600 VSA.

3. If AverageType is set to OFF, only one subframe is detected, demodulated, and analyzed.

   If AverageType is set to RMS (Video), after the first subframe is analyzed the signal segment corresponding to it is discarded and new signal samples are collected from the input to fill in the 10 msec-long signal buffer. When the buffer is full again a new subframe is detected, demodulated, and analyzed. These steps are repeated until SubframesToAverage subframes are processed.
Measurements

If for any reason a subframe is mis-detected the results from its analysis are discarded. The EVM results obtained from all the successfully detected, demodulated, and analyzed subframes are averaged to give the final averaged results. The EVM results from each successfully analyzed subframe are also recorded (in the variables without the Avg_prefix in their name).

4. The ChipRate parameter can be used to set the chip rate for the demodulation. Although the TD-SCDMA standard defines the chip rate to be 1.28 MHz, this parameter allows the user to enter nonstandard chip rates for test and analysis purposes. Of course, in order for the demodulation to be successful, the value of the ChipRate parameter must match the actual chip rate of the input signal.

5. The Alpha parameter can be used to set the measurement filter (root-raised cosine) alpha factor. Although the TD-SCDMA standard defines alpha to be 0.22, this parameter allows the user to enter nonstandard alpha values for test and analysis purposes. However, in order to get correct EVM results, the value of the Alpha parameter must match the alpha value used to generate the input signal.

6. The MirrorFrequencySpectrum parameter can be used to conjugate the input signal (when MirrorFrequencySpectrum is set to YES) before any other processing is done. Conjugating the input signal is necessary if the configuration of the mixers in your system has resulted in a conjugated signal compared to the one at the input of the up-converter. In this case, if MirrorFrequencySpectrum is not set to YES, the demodulation will fail.

7. The ActiveSlotThreshold parameter sets the active slot detection threshold, that is the power level (in dB with respect to the power level of the slot with the largest measured power) below which a slot will be considered as inactive.

8. Table 3-1 gives TD-SCDMA standard compliant allocations for downlink pilot, uplink pilot, scrambling, and basic midamble codes for the different code groups.
• The DownlinkPilotCode parameter sets the downlink pilot synchronization ID sequence (SYNC-DL). Downlink pilot synchronization (DwPTS) is used for DL synchronization and cell initial search. There are 32 different SYNC-DL code groups, which are used to distinguish base stations.

• The UplinkPilotCode parameter sets the uplink pilot synchronization ID sequence (SYNC-UL). Uplink pilot synchronization (UpPTS) is used for UL initial synchronization, random access and measurement for adjacent cell handoff. There are 256 different SYNC-UL codes, which can be divided into 32 groups. Each group includes 8 different SYNC-UL codes, i.e., each base station has 8 different SYNC-UL codes.

For test and analysis purposes UplinkPilotCode can be set to non-standard-compliant values (that do not follow the allocation scheme given in Table 3-1). However, in this case a warning message is displayed to remind the user that the value used is non-compliant.

• The ScrambleCode parameter sets the scramble code ID. There are 128 different scrambling codes, which are associated with a corresponding basic midamble code. Scrambling codes are cell specific and are used to identify separate cells.

<table>
<thead>
<tr>
<th>Group</th>
<th>DownlinkPilotCode</th>
<th>UplinkPilotCode</th>
<th>ScrambleCode</th>
<th>BasicMidambleID</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0-7</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>8-15</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>32</td>
<td>31</td>
<td>248-255</td>
<td>124</td>
<td>124</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>125</td>
<td>125</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>126</td>
<td>126</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>127</td>
<td>127</td>
</tr>
</tbody>
</table>

For test and analysis purposes UplinkPilotCode can be set to non-standard-compliant values (that do not follow the allocation scheme given in Table 3-1). However, in this case a warning message is displayed to remind the user that the value used is non-compliant.
For test and analysis purposes ScrambleCode can be set to non-standard-compliant values (that do not follow the allocation scheme given in Table 3-1). However, in this case a warning message is displayed to remind the user that the value used is non-compliant.

- The BasicMidambleID parameter sets the basic midamble code ID. The basic midamble code ID is used as training sequences for uplink and downlink channel estimation, power measurements and maintaining uplink synchronization. There are 128 different sequences divided into 32 groups corresponding to 32 SYNC-DL codes. Each group consists of 4 different basic midamble sequences, i.e. each base station has 4 different midambles.

For test and analysis purposes BasicMidambleID can be set to non-standard-compliant values (that do not follow the allocation scheme given in Table 3-1). However, in this case a warning message is displayed to remind the user that the value used is non-compliant.

9. The TrafficTimeslotMaxUsers parameter sets the maximum number of users in each timeslot (TS0 - TS6). This parameter is an array with 7 elements. If the number of elements specified is not exactly 7, the simulation will error out. Each array element must be an even number greater than or equal to 2 and smaller than or equal to 16.

10. The DespreadCodeLength and DespreadCodeChannel parameters can be used to specify the active code layer and channel for which channel EVM results will be provided.

11. The AnalysisTimeslot parameter can be used to specify which timeslot in the detected subframe will be analyzed. The available options are: timeslots 0 through 6 (TS0 - TS6), DwPTS, and UpPTS. When DwPTS or UpPTS is selected the results do not include any channel specific measurements (variables whose name starts with Ch or Avg_Ch), which means that the values of the DespreadCodeLength and DespreadCodeChannel parameters are ignored (not used).
TDSCDMA_FrameSync

Description  Synchronized frame generator
Library  TDSCDMA, Measurements
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>AnalysisTimeslot</td>
<td>timeslot to be analyzed: TS0, TS1, TS2, TS3, TS4, TS5, TS6</td>
<td>TS2</td>
<td>enum</td>
<td></td>
</tr>
<tr>
<td>SamplesPerSymbol</td>
<td>samples per symbol</td>
<td>8</td>
<td>int</td>
<td>[1, 32]</td>
</tr>
<tr>
<td>SyncCodeUsed</td>
<td>Code used in synchronization: DwPTS, UpPTS, Midamble</td>
<td>DwPTS</td>
<td>enum</td>
<td></td>
</tr>
<tr>
<td>SyncCodeIdx</td>
<td>index of basic synchronization code</td>
<td>0</td>
<td>int</td>
<td>[0, 31] when SyncCodeUsed = DwPTS; [0, 255] when SyncCodeUsed = UpPTS</td>
</tr>
<tr>
<td>ModPhase</td>
<td>type of modulation quadruples, valid only in downlink: S1, S2</td>
<td>S1</td>
<td>enum</td>
<td></td>
</tr>
<tr>
<td>MidambleAllocScheme</td>
<td>midamble allocation scheme: UE_Specific, Common, Default</td>
<td>Default</td>
<td>enum</td>
<td></td>
</tr>
<tr>
<td>BasicMidambleID</td>
<td>index of basic midamble</td>
<td>0</td>
<td>int</td>
<td>[0, 127]</td>
</tr>
<tr>
<td>K</td>
<td>maximum number of midamble shifts in a cell</td>
<td>16</td>
<td>int</td>
<td>{2, 4, 6, 8, 10, 12, 14, 16}</td>
</tr>
<tr>
<td>MidambleID</td>
<td>index of midamble</td>
<td>1</td>
<td>int</td>
<td>[1, K]</td>
</tr>
<tr>
<td>SpreadFactor</td>
<td>spreading factor</td>
<td>16</td>
<td>int</td>
<td>[1, 2, 4, 8, 16]</td>
</tr>
<tr>
<td>SpreadCode</td>
<td>index of spread code</td>
<td>1</td>
<td>int</td>
<td>[1, SpreadFactor]</td>
</tr>
<tr>
<td>PhyCHNum</td>
<td>number of channelization codes used in a timeslot</td>
<td>1</td>
<td>int</td>
<td>[1, 16]</td>
</tr>
</tbody>
</table>
Measurements

**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>DataI</td>
<td>input data</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>DataO</td>
<td>output data</td>
<td>complex</td>
</tr>
</tbody>
</table>

**Notes/Equations**

1. This subnetwork is used to synchronize the TDSCDMA signal frame-by-frame using pilot code or midamble code to find the first sample of the first frame and align the signal to the subframe boundary. Each firing, the subnetwork will consume data in one subframe length, buffer data in two subframe lengths, and produce a synchronized signal in one subframe length. The schematic for this subnetwork is shown in Figure 3-2.

![Figure 3-2. TDSCDMA_RF_CCDF Schematic](image-url)
2. The TDSCDMA input signal delay is introduced by a filter or device under test. The maximum delay that can be detected by this subnetwork is the length of one subframe. This model introduces an additional one-subframe delay that is padded with all zeros.

3. Synchronization is achieved by correlating the signals with the pilot codes or the midamble codes depending on the SyncCodeUsed setting. The largest correlation value is used to determine the synchronization point. Because the position of the pilot codes and midamble codes of a specific timeslot is fixed in each subframe, the frame boundary is easily determined.

4. SyncCodeUsed specifies the synchronization code.
   - DwPTS (downlink pilot codes) SyncCodeIdx and ModPhase synchronization-code-related parameters must be set.
   - UpPTS (uplink pilot codes) SyncCodeIdx synchronization-code-related parameter must be set.
   - Midamble (midamble codes) MidambleAllocScheme, BasicMidambleID, K, MidambleID, PhyChNum, SpreadFactor and SpreadCode synchronization-code-related parameters must be set.

5. AnalysisTimeslot determines the frame boundary when the maximum correlated value is found; set AnalysisTimeslot only if SyncCodeUsed is set to Midamble.

References

TDSCDMA_FwdChannel

Description  Multipath fading channel for forward link
Library  TDSCDMA, Measurements
Required Licenses

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Default</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Case</td>
<td>propagation conditions for multipath fading environments: case_1, case_2, case_3</td>
<td>case_1</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 data</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>Output</td>
<td>data after fading channel</td>
<td>timed</td>
</tr>
</tbody>
</table>

Notes/Equations

1. This subnetwork is used to simulate propagation conditions for multipath fading environments.

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

Each firing, 1 Output token is produced when 1 Input is consumed.
Reference

Measurements

**TDSCDMA_RF_CCDF**

**Description**  RF signal complementary cumulative distribution function

**Library**  TDSCDMA, Measurements

**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>RLoad</td>
<td>reference resistance</td>
<td>DefaultRIn</td>
<td>Ohm</td>
<td>real</td>
<td>(0, ∞)</td>
</tr>
<tr>
<td>RTemp</td>
<td>temperature of reference resistor, in degrees C</td>
<td>DefaultRTemp</td>
<td>real</td>
<td>[-273.15, ∞)</td>
<td></td>
</tr>
<tr>
<td>SamplesPerSymbol</td>
<td>samples per symbol</td>
<td>8</td>
<td>int</td>
<td>[1, 32]</td>
<td></td>
</tr>
<tr>
<td>SlotIndex</td>
<td>index of slot</td>
<td>2</td>
<td>int</td>
<td>[0, 6]</td>
<td></td>
</tr>
<tr>
<td>NumSlotsMeasured</td>
<td>number of slot to be measured</td>
<td>5</td>
<td>int</td>
<td>[1, 300]</td>
<td></td>
</tr>
<tr>
<td>OutputPoint</td>
<td>indicate output precision</td>
<td>100</td>
<td>int</td>
<td>[3, 100]</td>
<td></td>
</tr>
<tr>
<td>SystemDelay</td>
<td>delay due to filters</td>
<td>64</td>
<td>int</td>
<td>(0, ∞)</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>in</td>
<td>input signals</td>
<td>timed</td>
</tr>
</tbody>
</table>

**Notes/Equations**

1. This subnetwork measures the complementary cumulative distribution function (CCDF) of the RF signal.

   The schematic for this subnetwork is shown in Figure 3-4.
2. TDSCDMA_RF_CCDF measures the distribution function according to input signal power; results are collected by four NumericSink models. The distribution range is sent to the SignalRange NumericSink; here the distribution range is divided into segments (based on the OutputPoint setting). Corresponding distribution probabilities are measured on these segments and sent to the CCDF NumericSink.

NumericSinks PeakPower, MeanPower and SignalRange units are dBm

3. SlotIndex indicates which slot in a frame will be measured; CCDF can be measured on several time slots. The slots with No. SlotIndex in NumSlotsMeasured consecutive subframes are combined to get more precise results.

4. The signal is regarded as subframe-synchronized. SystemDelay indicates the number of delay in samples caused by filters and other devices. If the delay is not a multiple of subframe, extra delay will be added in DelayRF so that the test begins at the first effective data.

References

**Measurements**

**TDSCDMA_RF_PwrMeasure**

**Description**  RF power meter  
**Library**  TDSCDMA, Measurements  
**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>RLoad</td>
<td>reference resistance</td>
<td>DefaultRln</td>
<td>Ohm</td>
<td>real</td>
<td>(0, ∞)</td>
</tr>
<tr>
<td>RTemp</td>
<td>temperature of reference resistor, in degrees C</td>
<td>DefaultRTemp</td>
<td>real</td>
<td>(0, 273.15, ∞)</td>
<td></td>
</tr>
<tr>
<td>FCarrier</td>
<td>carrier frequency</td>
<td>1900MHz</td>
<td>Hz</td>
<td>real</td>
<td>(0, ∞)</td>
</tr>
<tr>
<td>AnalysisTimeslot</td>
<td>timeslot to be analyzed: TS0, TS1, TS2, TS3, TS4, TS5, TS6</td>
<td>TS2</td>
<td>enum</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SamplesPerSymbol</td>
<td>samples per symbol</td>
<td>8</td>
<td>int</td>
<td></td>
<td>[1, 32]</td>
</tr>
<tr>
<td>FilterLength</td>
<td>length of raised cosine filters in number of symbols</td>
<td>16</td>
<td>int</td>
<td></td>
<td>(0, ∞)</td>
</tr>
<tr>
<td>NumSlotsMeasured</td>
<td>number of slot to be measured</td>
<td>3</td>
<td>int</td>
<td></td>
<td>[1, 300]</td>
</tr>
<tr>
<td>SyncCodeUsed</td>
<td>Code used in synchronization: DwPTS, UpPTS, Midamble</td>
<td>DwPTS</td>
<td>enum</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SyncCodeIdx</td>
<td>index of basic synchronization code</td>
<td>0</td>
<td>int</td>
<td></td>
<td>[0, 31] when SyncCodeUsed = DwPTS; [0, 255] when SyncCodeUsed = UpPTS</td>
</tr>
<tr>
<td>ModPhase</td>
<td>type of modulation quadruples, valid only in downlink: S1, S2</td>
<td>S1</td>
<td>enum</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MidambleAllocScheme</td>
<td>midamble allocation scheme: UE_Specific, Common, Default</td>
<td>Default</td>
<td>enum</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BasicMidambleID</td>
<td>index of basic midamble</td>
<td>0</td>
<td>int</td>
<td></td>
<td>[0, 127]</td>
</tr>
</tbody>
</table>
This subnetwork measures the average power and power vs. time of the RF signal.

The schematic for this subnetwork is shown in Figure 3-5.

There are two outputs:

- One output is the average power for each time slot GP, DwPTS and UpPTS. Ten values will be fed into NumericSink AverageTotalPower (see Figure 3-5). They are average power for Slot 0, DwPTS, GP, UpPTS, Slot 1 to Slot 6 sequentially. The power of one slot can be averaged with correspondent slots in NumSlotsMeasured subframes. For example, if NumSlotsMeasured is 8 the average power of Slot 1 will be the average power of Slot 1 in all 8 subframes. Note that the GP part in each slot will not be counted when measuring the average power.

- One output is the average power of each chip in one subframe. 6400 values will be fed into NumericSink PowerVsTime (see Figure 3-5). The power of each chip will be averaged with correspondent chips in NumSlotsMeasured subframes.

### 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 signals</td>
<td>timed</td>
</tr>
</tbody>
</table>

### Notes/Equations

1. This subnetwork measures the average power and power vs. time of the RF signal.

The schematic for this subnetwork is shown in Figure 3-5.

2. There are two outputs:

- One output is the average power for each time slot GP, DwPTS and UpPTS. Ten values will be fed into NumericSink AverageTotalPower (see Figure 3-5). They are average power for Slot 0, DwPTS, GP, UpPTS, Slot 1 to Slot 6 sequentially. The power of one slot can be averaged with correspondent slots in NumSlotsMeasured subframes. For example, if NumSlotsMeasured is 8 the average power of Slot 1 will be the average power of Slot 1 in all 8 subframes. Note that the GP part in each slot will not be counted when measuring the average power.

- One output is the average power of each chip in one subframe. 6400 values will be fed into NumericSink PowerVsTime (see Figure 3-5). The power of each chip will be averaged with correspondent chips in NumSlotsMeasured subframes.
3. NumericSinks PowerVsTime and AverageTotalPower units are dBm.

4. SyncCodeUsed specifies the synchronization code.
   - DwPTS (downlink pilot codes)  SyncCodeIdx and ModPhase
     synchronization-code-related parameters must be set.
   - UpPTS (uplink pilot codes)  SyncCodeIdx synchronization-code-related
     parameter must be set.
   - Midamble (midamble codes)  MidambleAllocScheme, BasicMidambleID, K,
     MidambleID, PhyChNum, SpreadFactor and SpreadCode
     synchronization-code-related parameters must be set.

5. AnalysisTimeslot determines the frame boundary when the maximum
   correlated value is found; set AnalysisTimeslot only if SyncCodeUsed is set to
   Midamble.

References

    Group Radio Access Network; Physical channels and mapping of transport
    channels onto physical channels onto physical channels (TDD) (Release 4),
    version 4.3.0, Dec., 2001
TDSCDMA_RevChannel

Description  Multipath fading channel for reverse link
Library   TDSCDMA, Measurements
Required Licenses

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Default</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Case</td>
<td>propagation conditions for multipath fading environments: case_1, case_2, case_3</td>
<td>case_1</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 data</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>Output</td>
<td>data after fading channel</td>
<td>timed</td>
</tr>
</tbody>
</table>

Notes/Equations

1. This subnetwork is used to simulate propagation conditions for multipath fading environments.
2. The schematic for this subnetwork is shown in Figure 3-6.
3. Each firing, 1 Output token is produced when 1 Input is consumed.
Measurements

Figure 3-6. Schematic of TDSCDMA_RevChannel

References

Chapter 4: Modems
TDSCDMA_BurstDeMux

Description  Burst demultiplexer  
Library  TDSCDMA, Modems  
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>K</td>
<td>maximum number of midamble shifts in a cell</td>
<td>16</td>
<td>int</td>
<td>{2, 4, 6, 8, 10, 12, 14, 16}</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</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>data output</td>
<td>complex</td>
</tr>
<tr>
<td>3</td>
<td>mid</td>
<td>midamble output</td>
<td>complex</td>
</tr>
</tbody>
</table>

Notes/Equations

1. This subnetwork is used to demultiplex data and midamble from a burst.

The schematic for this subnetwork is shown in Figure 4-1.

Each firing, 864 tokens are consumed while \((352+W) \times 2\) data tokens and 144 midamble tokens are produced, where \(W=128/K\), which is the channel estimation window length.
The burst structure is illustrated in Figure 4-2.

Figure 4-2. Burst Structure of Traffic Burst Format

References

TDSCDMA_BurstMux

Description  Burst multiplexer
Library  TDSCDMA, Modems
Class  SDFTDSCDMA_BurstMux
Required Licenses

<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</td>
<td>complex</td>
</tr>
<tr>
<td>2</td>
<td>Midamble</td>
<td>midamble</td>
<td>complex</td>
</tr>
</tbody>
</table>

<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>burst</td>
<td>complex</td>
</tr>
</tbody>
</table>

Notes/Equations
1. This model multiplexes data and midamble and forms a burst.
   Each firing, 864 Output tokens are produced when 704 Data tokens and 144 Midamble tokens are consumed.
2. The burst structure of the traffic burst format is illustrated in Figure 4-3.
Figure 4-3. Burst Structure of Traffic Burst Format

References

TDSCDMA_DPCH_DataDeMux

Description  DeMultiplexer for data, TFCI, SS, and TPC in DPCH
Library  TDSCDMA, Modems

Required Licenses

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Default</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>BitsPerSlot</td>
<td>number of bits per slot</td>
<td>88</td>
<td>int</td>
</tr>
<tr>
<td>N_Data1</td>
<td>number of data bits in the first data field</td>
<td>44</td>
<td>int</td>
</tr>
<tr>
<td>N_Data2</td>
<td>number of data bits in the second data field</td>
<td>44</td>
<td>int</td>
</tr>
<tr>
<td>N_TFCI1</td>
<td>number of TFCI bits in the first TFCI field</td>
<td>0</td>
<td>int</td>
</tr>
<tr>
<td>N_TFCI2</td>
<td>number of TFCI bits in the second TFCI field</td>
<td>0</td>
<td>int</td>
</tr>
<tr>
<td>N_SS</td>
<td>number of SS bits in the slot</td>
<td>0</td>
<td>int</td>
</tr>
<tr>
<td>N_TPC</td>
<td>number of TPC bits in the slot</td>
<td>0</td>
<td>int</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>Data</td>
<td>data of dedicated physical channel</td>
<td>real</td>
</tr>
<tr>
<td>3</td>
<td>TFCI</td>
<td>transport format combination indicator bits</td>
<td>real</td>
</tr>
<tr>
<td>4</td>
<td>SS</td>
<td>information bits for uplink synchronization control</td>
<td>real</td>
</tr>
<tr>
<td>5</td>
<td>TPC</td>
<td>transmit power control bits</td>
<td>real</td>
</tr>
</tbody>
</table>
1. This subnetwork is used to demultiplex data, TFCI bits, SS bits and TPC bits from a DPCH.

   The schematic for this subnetwork is shown in Figure 4-4.

2. The structure of a typical slot is illustrated in Figure 4-5.
3. Time slot formats for the downlink with QPSK modulation are given in Table 4-1; time slot formats for uplink with QPSK modulation are given in Table 4-2; time slot formats for both links with 8PSK modulation are given in Table 4-3.

Table 4-1. Downlink Time Slot Formats

<table>
<thead>
<tr>
<th>Slot Format #</th>
<th>Spread Factor</th>
<th>Midamble Length (chips)</th>
<th>NTFCI Code Word (bits)</th>
<th>NSS &amp; NTPC (bits)</th>
<th>Bits/Slot</th>
<th>NData/Slot (bits)</th>
<th>Ndata/Data Field (1) (bits)</th>
<th>Ndata/Data Field (2) (bits)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>16</td>
<td>144</td>
<td>0</td>
<td>0 &amp; 0</td>
<td>88</td>
<td>88</td>
<td>44</td>
<td>44</td>
</tr>
<tr>
<td>1</td>
<td>16</td>
<td>144</td>
<td>4</td>
<td>0 &amp; 0</td>
<td>88</td>
<td>86</td>
<td>42</td>
<td>44</td>
</tr>
<tr>
<td>2</td>
<td>16</td>
<td>144</td>
<td>8</td>
<td>0 &amp; 0</td>
<td>88</td>
<td>84</td>
<td>42</td>
<td>42</td>
</tr>
<tr>
<td>3</td>
<td>16</td>
<td>144</td>
<td>16</td>
<td>0 &amp; 0</td>
<td>88</td>
<td>80</td>
<td>40</td>
<td>40</td>
</tr>
<tr>
<td>4</td>
<td>16</td>
<td>144</td>
<td>32</td>
<td>0 &amp; 0</td>
<td>88</td>
<td>72</td>
<td>36</td>
<td>36</td>
</tr>
<tr>
<td>5</td>
<td>16</td>
<td>144</td>
<td>0</td>
<td>2 &amp; 2</td>
<td>88</td>
<td>84</td>
<td>44</td>
<td>40</td>
</tr>
<tr>
<td>6</td>
<td>16</td>
<td>144</td>
<td>4</td>
<td>2 &amp; 2</td>
<td>88</td>
<td>82</td>
<td>42</td>
<td>40</td>
</tr>
<tr>
<td>7</td>
<td>16</td>
<td>144</td>
<td>8</td>
<td>2 &amp; 2</td>
<td>88</td>
<td>80</td>
<td>42</td>
<td>38</td>
</tr>
<tr>
<td>8</td>
<td>16</td>
<td>144</td>
<td>16</td>
<td>2 &amp; 2</td>
<td>88</td>
<td>76</td>
<td>40</td>
<td>36</td>
</tr>
<tr>
<td>9</td>
<td>16</td>
<td>144</td>
<td>32</td>
<td>2 &amp; 2</td>
<td>88</td>
<td>68</td>
<td>36</td>
<td>32</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>144</td>
<td>0</td>
<td>0 &amp; 0</td>
<td>1408</td>
<td>1408</td>
<td>704</td>
<td>704</td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>144</td>
<td>4</td>
<td>0 &amp; 0</td>
<td>1408</td>
<td>1406</td>
<td>702</td>
<td>704</td>
</tr>
<tr>
<td>12</td>
<td>1</td>
<td>144</td>
<td>8</td>
<td>0 &amp; 0</td>
<td>1408</td>
<td>1404</td>
<td>702</td>
<td>702</td>
</tr>
<tr>
<td>13</td>
<td>1</td>
<td>144</td>
<td>16</td>
<td>0 &amp; 0</td>
<td>1408</td>
<td>1400</td>
<td>700</td>
<td>700</td>
</tr>
<tr>
<td>14</td>
<td>1</td>
<td>144</td>
<td>32</td>
<td>0 &amp; 0</td>
<td>1408</td>
<td>1392</td>
<td>696</td>
<td>696</td>
</tr>
<tr>
<td>15</td>
<td>1</td>
<td>144</td>
<td>0</td>
<td>2 &amp; 2</td>
<td>1408</td>
<td>1404</td>
<td>704</td>
<td>700</td>
</tr>
<tr>
<td>16</td>
<td>1</td>
<td>144</td>
<td>4</td>
<td>2 &amp; 2</td>
<td>1408</td>
<td>1402</td>
<td>702</td>
<td>700</td>
</tr>
<tr>
<td>17</td>
<td>1</td>
<td>144</td>
<td>8</td>
<td>2 &amp; 2</td>
<td>1408</td>
<td>1400</td>
<td>702</td>
<td>698</td>
</tr>
<tr>
<td>18</td>
<td>1</td>
<td>144</td>
<td>16</td>
<td>2 &amp; 2</td>
<td>1408</td>
<td>1396</td>
<td>696</td>
<td>696</td>
</tr>
<tr>
<td>19</td>
<td>1</td>
<td>144</td>
<td>32</td>
<td>2 &amp; 2</td>
<td>1408</td>
<td>1388</td>
<td>696</td>
<td>692</td>
</tr>
<tr>
<td>20</td>
<td>1</td>
<td>144</td>
<td>0</td>
<td>32 &amp; 32</td>
<td>1408</td>
<td>1344</td>
<td>704</td>
<td>640</td>
</tr>
<tr>
<td>21</td>
<td>1</td>
<td>144</td>
<td>4</td>
<td>32 &amp; 32</td>
<td>1408</td>
<td>1342</td>
<td>702</td>
<td>640</td>
</tr>
<tr>
<td>22</td>
<td>1</td>
<td>144</td>
<td>8</td>
<td>32 &amp; 32</td>
<td>1408</td>
<td>1340</td>
<td>702</td>
<td>638</td>
</tr>
<tr>
<td>23</td>
<td>1</td>
<td>144</td>
<td>16</td>
<td>32 &amp; 32</td>
<td>1408</td>
<td>1336</td>
<td>700</td>
<td>636</td>
</tr>
<tr>
<td>24</td>
<td>1</td>
<td>144</td>
<td>32</td>
<td>32 &amp; 32</td>
<td>1408</td>
<td>1328</td>
<td>696</td>
<td>632</td>
</tr>
<tr>
<td>Slot</td>
<td>Spread Factor</td>
<td>Midamble length (chips)</td>
<td>NTFCI code word (bits)</td>
<td>NSS &amp; NTPC (bits)</td>
<td>Bits/ slot</td>
<td>NData/ Slot (bits)</td>
<td>Ndata/ data field (1) (bits)</td>
<td>Ndata/ data field (2) (bits)</td>
</tr>
<tr>
<td>------</td>
<td>---------------</td>
<td>------------------------</td>
<td>------------------------</td>
<td>-------------------</td>
<td>------------</td>
<td>-------------------</td>
<td>-------------------------------</td>
<td>-------------------------------</td>
</tr>
<tr>
<td>0</td>
<td>16</td>
<td>144</td>
<td>0</td>
<td>0 &amp; 0</td>
<td>88</td>
<td>88</td>
<td>44</td>
<td>44</td>
</tr>
<tr>
<td>1</td>
<td>16</td>
<td>144</td>
<td>4</td>
<td>0 &amp; 0</td>
<td>88</td>
<td>86</td>
<td>42</td>
<td>44</td>
</tr>
<tr>
<td>2</td>
<td>16</td>
<td>144</td>
<td>8</td>
<td>0 &amp; 0</td>
<td>88</td>
<td>84</td>
<td>42</td>
<td>42</td>
</tr>
<tr>
<td>3</td>
<td>16</td>
<td>144</td>
<td>16</td>
<td>0 &amp; 0</td>
<td>88</td>
<td>80</td>
<td>40</td>
<td>40</td>
</tr>
<tr>
<td>4</td>
<td>16</td>
<td>144</td>
<td>32</td>
<td>0 &amp; 0</td>
<td>88</td>
<td>72</td>
<td>36</td>
<td>36</td>
</tr>
<tr>
<td>5</td>
<td>16</td>
<td>144</td>
<td>0</td>
<td>2 &amp; 2</td>
<td>88</td>
<td>84</td>
<td>44</td>
<td>40</td>
</tr>
<tr>
<td>6</td>
<td>16</td>
<td>144</td>
<td>4</td>
<td>2 &amp; 2</td>
<td>88</td>
<td>82</td>
<td>42</td>
<td>40</td>
</tr>
<tr>
<td>7</td>
<td>16</td>
<td>144</td>
<td>8</td>
<td>2 &amp; 2</td>
<td>88</td>
<td>80</td>
<td>42</td>
<td>38</td>
</tr>
<tr>
<td>8</td>
<td>16</td>
<td>144</td>
<td>16</td>
<td>2 &amp; 2</td>
<td>88</td>
<td>76</td>
<td>40</td>
<td>36</td>
</tr>
<tr>
<td>9</td>
<td>16</td>
<td>144</td>
<td>32</td>
<td>2 &amp; 2</td>
<td>88</td>
<td>68</td>
<td>36</td>
<td>32</td>
</tr>
<tr>
<td>10</td>
<td>8</td>
<td>144</td>
<td>0</td>
<td>0 &amp; 0</td>
<td>176</td>
<td>176</td>
<td>88</td>
<td>88</td>
</tr>
<tr>
<td>11</td>
<td>8</td>
<td>144</td>
<td>4</td>
<td>0 &amp; 0</td>
<td>176</td>
<td>174</td>
<td>86</td>
<td>88</td>
</tr>
<tr>
<td>12</td>
<td>8</td>
<td>144</td>
<td>8</td>
<td>0 &amp; 0</td>
<td>176</td>
<td>172</td>
<td>86</td>
<td>86</td>
</tr>
<tr>
<td>13</td>
<td>8</td>
<td>144</td>
<td>16</td>
<td>0 &amp; 0</td>
<td>176</td>
<td>168</td>
<td>84</td>
<td>84</td>
</tr>
<tr>
<td>14</td>
<td>8</td>
<td>144</td>
<td>32</td>
<td>0 &amp; 0</td>
<td>176</td>
<td>160</td>
<td>80</td>
<td>80</td>
</tr>
<tr>
<td>15</td>
<td>8</td>
<td>144</td>
<td>0</td>
<td>2 &amp; 2</td>
<td>176</td>
<td>176</td>
<td>88</td>
<td>84</td>
</tr>
<tr>
<td>16</td>
<td>8</td>
<td>144</td>
<td>4</td>
<td>2 &amp; 2</td>
<td>176</td>
<td>170</td>
<td>86</td>
<td>84</td>
</tr>
<tr>
<td>17</td>
<td>8</td>
<td>144</td>
<td>8</td>
<td>2 &amp; 2</td>
<td>176</td>
<td>168</td>
<td>86</td>
<td>82</td>
</tr>
<tr>
<td>18</td>
<td>8</td>
<td>144</td>
<td>16</td>
<td>2 &amp; 2</td>
<td>176</td>
<td>164</td>
<td>84</td>
<td>80</td>
</tr>
<tr>
<td>19</td>
<td>8</td>
<td>144</td>
<td>32</td>
<td>2 &amp; 2</td>
<td>176</td>
<td>156</td>
<td>80</td>
<td>76</td>
</tr>
<tr>
<td>20</td>
<td>8</td>
<td>144</td>
<td>0</td>
<td>4 &amp; 4</td>
<td>176</td>
<td>168</td>
<td>88</td>
<td>80</td>
</tr>
<tr>
<td>21</td>
<td>8</td>
<td>144</td>
<td>4</td>
<td>4 &amp; 4</td>
<td>176</td>
<td>166</td>
<td>86</td>
<td>80</td>
</tr>
<tr>
<td>22</td>
<td>8</td>
<td>144</td>
<td>8</td>
<td>4 &amp; 4</td>
<td>176</td>
<td>164</td>
<td>86</td>
<td>78</td>
</tr>
<tr>
<td>23</td>
<td>8</td>
<td>144</td>
<td>16</td>
<td>4 &amp; 4</td>
<td>176</td>
<td>160</td>
<td>84</td>
<td>76</td>
</tr>
<tr>
<td>24</td>
<td>8</td>
<td>144</td>
<td>32</td>
<td>4 &amp; 4</td>
<td>176</td>
<td>152</td>
<td>80</td>
<td>72</td>
</tr>
<tr>
<td>25</td>
<td>4</td>
<td>144</td>
<td>0</td>
<td>0 &amp; 0</td>
<td>352</td>
<td>352</td>
<td>176</td>
<td>176</td>
</tr>
<tr>
<td>26</td>
<td>4</td>
<td>144</td>
<td>4</td>
<td>0 &amp; 0</td>
<td>352</td>
<td>350</td>
<td>174</td>
<td>176</td>
</tr>
<tr>
<td>27</td>
<td>4</td>
<td>144</td>
<td>8</td>
<td>0 &amp; 0</td>
<td>352</td>
<td>348</td>
<td>174</td>
<td>174</td>
</tr>
<tr>
<td>28</td>
<td>4</td>
<td>144</td>
<td>16</td>
<td>0 &amp; 0</td>
<td>352</td>
<td>344</td>
<td>172</td>
<td>172</td>
</tr>
<tr>
<td>29</td>
<td>4</td>
<td>144</td>
<td>32</td>
<td>0 &amp; 0</td>
<td>352</td>
<td>336</td>
<td>168</td>
<td>168</td>
</tr>
<tr>
<td>30</td>
<td>4</td>
<td>144</td>
<td>0</td>
<td>2 &amp; 2</td>
<td>352</td>
<td>348</td>
<td>176</td>
<td>172</td>
</tr>
<tr>
<td>31</td>
<td>4</td>
<td>144</td>
<td>4</td>
<td>2 &amp; 2</td>
<td>352</td>
<td>346</td>
<td>174</td>
<td>172</td>
</tr>
</tbody>
</table>
### Table 4-2. Uplink Time Slot Formats

<table>
<thead>
<tr>
<th>Slot Format</th>
<th>Spread Factor</th>
<th>Midamble length (chips)</th>
<th>NTFCI code word (bits)</th>
<th>NSS &amp; NTPC (bits)</th>
<th>Bits/slot</th>
<th>NData/Slot (bits)</th>
<th>Ndata/data field (1) (bits)</th>
<th>Ndata/data field (2) (bits)</th>
</tr>
</thead>
<tbody>
<tr>
<td>32</td>
<td>4</td>
<td>144</td>
<td>8</td>
<td>2 &amp; 2</td>
<td>352</td>
<td>344</td>
<td>174</td>
<td>170</td>
</tr>
<tr>
<td>33</td>
<td>4</td>
<td>144</td>
<td>16</td>
<td>2 &amp; 2</td>
<td>352</td>
<td>340</td>
<td>172</td>
<td>168</td>
</tr>
<tr>
<td>34</td>
<td>4</td>
<td>144</td>
<td>32</td>
<td>2 &amp; 2</td>
<td>352</td>
<td>332</td>
<td>168</td>
<td>164</td>
</tr>
<tr>
<td>35</td>
<td>4</td>
<td>144</td>
<td>0</td>
<td>8 &amp; 8</td>
<td>352</td>
<td>336</td>
<td>176</td>
<td>160</td>
</tr>
<tr>
<td>36</td>
<td>4</td>
<td>144</td>
<td>4</td>
<td>8 &amp; 8</td>
<td>352</td>
<td>334</td>
<td>174</td>
<td>160</td>
</tr>
<tr>
<td>37</td>
<td>4</td>
<td>144</td>
<td>8</td>
<td>8 &amp; 8</td>
<td>352</td>
<td>332</td>
<td>174</td>
<td>158</td>
</tr>
<tr>
<td>38</td>
<td>4</td>
<td>144</td>
<td>16</td>
<td>8 &amp; 8</td>
<td>352</td>
<td>328</td>
<td>172</td>
<td>156</td>
</tr>
<tr>
<td>39</td>
<td>4</td>
<td>144</td>
<td>32</td>
<td>8 &amp; 8</td>
<td>352</td>
<td>320</td>
<td>168</td>
<td>152</td>
</tr>
<tr>
<td>40</td>
<td>2</td>
<td>144</td>
<td>0</td>
<td>0 &amp; 0</td>
<td>704</td>
<td>704</td>
<td>352</td>
<td>352</td>
</tr>
<tr>
<td>41</td>
<td>2</td>
<td>144</td>
<td>4</td>
<td>0 &amp; 0</td>
<td>704</td>
<td>702</td>
<td>350</td>
<td>352</td>
</tr>
<tr>
<td>42</td>
<td>2</td>
<td>144</td>
<td>8</td>
<td>0 &amp; 0</td>
<td>704</td>
<td>700</td>
<td>350</td>
<td>350</td>
</tr>
<tr>
<td>43</td>
<td>2</td>
<td>144</td>
<td>16</td>
<td>0 &amp; 0</td>
<td>704</td>
<td>696</td>
<td>348</td>
<td>348</td>
</tr>
<tr>
<td>44</td>
<td>2</td>
<td>144</td>
<td>32</td>
<td>0 &amp; 0</td>
<td>704</td>
<td>688</td>
<td>344</td>
<td>344</td>
</tr>
<tr>
<td>45</td>
<td>2</td>
<td>144</td>
<td>0</td>
<td>2 &amp; 2</td>
<td>704</td>
<td>700</td>
<td>352</td>
<td>348</td>
</tr>
<tr>
<td>46</td>
<td>2</td>
<td>144</td>
<td>4</td>
<td>2 &amp; 2</td>
<td>704</td>
<td>698</td>
<td>350</td>
<td>348</td>
</tr>
<tr>
<td>47</td>
<td>2</td>
<td>144</td>
<td>8</td>
<td>2 &amp; 2</td>
<td>704</td>
<td>696</td>
<td>350</td>
<td>346</td>
</tr>
<tr>
<td>48</td>
<td>2</td>
<td>144</td>
<td>16</td>
<td>2 &amp; 2</td>
<td>704</td>
<td>692</td>
<td>348</td>
<td>344</td>
</tr>
<tr>
<td>49</td>
<td>2</td>
<td>144</td>
<td>32</td>
<td>2 &amp; 2</td>
<td>704</td>
<td>684</td>
<td>344</td>
<td>340</td>
</tr>
<tr>
<td>50</td>
<td>2</td>
<td>144</td>
<td>0</td>
<td>16 &amp; 16</td>
<td>704</td>
<td>672</td>
<td>352</td>
<td>320</td>
</tr>
<tr>
<td>51</td>
<td>2</td>
<td>144</td>
<td>4</td>
<td>16 &amp; 16</td>
<td>704</td>
<td>670</td>
<td>350</td>
<td>320</td>
</tr>
<tr>
<td>52</td>
<td>2</td>
<td>144</td>
<td>8</td>
<td>16 &amp; 16</td>
<td>704</td>
<td>668</td>
<td>350</td>
<td>318</td>
</tr>
<tr>
<td>53</td>
<td>2</td>
<td>144</td>
<td>16</td>
<td>16 &amp; 16</td>
<td>704</td>
<td>664</td>
<td>348</td>
<td>316</td>
</tr>
<tr>
<td>54</td>
<td>2</td>
<td>144</td>
<td>32</td>
<td>16 &amp; 16</td>
<td>704</td>
<td>656</td>
<td>344</td>
<td>312</td>
</tr>
<tr>
<td>55</td>
<td>1</td>
<td>144</td>
<td>0</td>
<td>0 &amp; 0</td>
<td>1408</td>
<td>1408</td>
<td>704</td>
<td>704</td>
</tr>
<tr>
<td>56</td>
<td>1</td>
<td>144</td>
<td>4</td>
<td>0 &amp; 0</td>
<td>1408</td>
<td>1406</td>
<td>702</td>
<td>704</td>
</tr>
<tr>
<td>57</td>
<td>1</td>
<td>144</td>
<td>8</td>
<td>0 &amp; 0</td>
<td>1408</td>
<td>1404</td>
<td>702</td>
<td>702</td>
</tr>
<tr>
<td>58</td>
<td>1</td>
<td>144</td>
<td>16</td>
<td>0 &amp; 0</td>
<td>1408</td>
<td>1400</td>
<td>700</td>
<td>700</td>
</tr>
<tr>
<td>59</td>
<td>1</td>
<td>144</td>
<td>32</td>
<td>0 &amp; 0</td>
<td>1408</td>
<td>1392</td>
<td>696</td>
<td>696</td>
</tr>
<tr>
<td>60</td>
<td>1</td>
<td>144</td>
<td>0</td>
<td>2 &amp; 2</td>
<td>1408</td>
<td>1404</td>
<td>704</td>
<td>700</td>
</tr>
<tr>
<td>61</td>
<td>1</td>
<td>144</td>
<td>4</td>
<td>2 &amp; 2</td>
<td>1408</td>
<td>1402</td>
<td>702</td>
<td>700</td>
</tr>
<tr>
<td>62</td>
<td>1</td>
<td>144</td>
<td>8</td>
<td>2 &amp; 2</td>
<td>1408</td>
<td>1400</td>
<td>702</td>
<td>698</td>
</tr>
<tr>
<td>63</td>
<td>1</td>
<td>144</td>
<td>16</td>
<td>2 &amp; 2</td>
<td>1408</td>
<td>1396</td>
<td>700</td>
<td>696</td>
</tr>
</tbody>
</table>
### Table 4-2. Uplink Time Slot Formats

<table>
<thead>
<tr>
<th>Slot Format</th>
<th>Spread Factor</th>
<th>Midamble length (chips)</th>
<th>NTFCI code word (bits)</th>
<th>NSS &amp; NTFCI code (bits)</th>
<th>Bits/Slot (bits)</th>
<th>NData/Slot (bits)</th>
<th>Ndata/data field (1) (bits)</th>
<th>Ndata/data field (2) (bits)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>144</td>
<td>0</td>
<td>0 &amp; 0</td>
<td>2112</td>
<td>2112</td>
<td>1056</td>
<td>1056</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>144</td>
<td>6</td>
<td>0 &amp; 0</td>
<td>2112</td>
<td>2109</td>
<td>1053</td>
<td>1056</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>144</td>
<td>12</td>
<td>0 &amp; 0</td>
<td>2112</td>
<td>2106</td>
<td>1053</td>
<td>1053</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>144</td>
<td>24</td>
<td>0 &amp; 0</td>
<td>2112</td>
<td>2100</td>
<td>1050</td>
<td>1050</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>144</td>
<td>48</td>
<td>0 &amp; 0</td>
<td>2112</td>
<td>2088</td>
<td>1044</td>
<td>1044</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>144</td>
<td>0</td>
<td>3 &amp; 3</td>
<td>2112</td>
<td>2106</td>
<td>1056</td>
<td>1050</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>144</td>
<td>6</td>
<td>3 &amp; 3</td>
<td>2112</td>
<td>2103</td>
<td>1053</td>
<td>1050</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>144</td>
<td>12</td>
<td>3 &amp; 3</td>
<td>2112</td>
<td>2100</td>
<td>1053</td>
<td>1047</td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>144</td>
<td>24</td>
<td>3 &amp; 3</td>
<td>2112</td>
<td>2094</td>
<td>1050</td>
<td>1044</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>144</td>
<td>48</td>
<td>3 &amp; 3</td>
<td>2112</td>
<td>2082</td>
<td>1044</td>
<td>1038</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>144</td>
<td>0</td>
<td>48 &amp; 48</td>
<td>2112</td>
<td>2016</td>
<td>1056</td>
<td>960</td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>144</td>
<td>6</td>
<td>48 &amp; 48</td>
<td>2112</td>
<td>2013</td>
<td>1053</td>
<td>960</td>
</tr>
<tr>
<td>12</td>
<td>1</td>
<td>144</td>
<td>12</td>
<td>48 &amp; 48</td>
<td>2112</td>
<td>2010</td>
<td>1053</td>
<td>957</td>
</tr>
<tr>
<td>13</td>
<td>1</td>
<td>144</td>
<td>24</td>
<td>48 &amp; 48</td>
<td>2112</td>
<td>2004</td>
<td>1050</td>
<td>954</td>
</tr>
<tr>
<td>14</td>
<td>1</td>
<td>144</td>
<td>48</td>
<td>48 &amp; 48</td>
<td>2112</td>
<td>1992</td>
<td>1044</td>
<td>948</td>
</tr>
<tr>
<td>15</td>
<td>16</td>
<td>144</td>
<td>0</td>
<td>0 &amp; 0</td>
<td>132</td>
<td>132</td>
<td>66</td>
<td>66</td>
</tr>
<tr>
<td>16</td>
<td>16</td>
<td>144</td>
<td>6</td>
<td>0 &amp; 0</td>
<td>132</td>
<td>129</td>
<td>63</td>
<td>66</td>
</tr>
<tr>
<td>17</td>
<td>16</td>
<td>144</td>
<td>12</td>
<td>0 &amp; 0</td>
<td>132</td>
<td>126</td>
<td>63</td>
<td>63</td>
</tr>
<tr>
<td>18</td>
<td>16</td>
<td>144</td>
<td>24</td>
<td>0 &amp; 0</td>
<td>132</td>
<td>120</td>
<td>60</td>
<td>60</td>
</tr>
<tr>
<td>19</td>
<td>16</td>
<td>144</td>
<td>48</td>
<td>0 &amp; 0</td>
<td>132</td>
<td>108</td>
<td>54</td>
<td>54</td>
</tr>
</tbody>
</table>
Modems

Table 4-3. 8PSK Modulation Time Slot Formats

<table>
<thead>
<tr>
<th>Slot Format</th>
<th>Spread Factor</th>
<th>Midamble length (chips)</th>
<th>NTFCI code word (bits)</th>
<th>NSS &amp; NTPC (bits)</th>
<th>Bits/Slot (bits)</th>
<th>NData/Slot (bits)</th>
<th>Ndata/data field (1) (bits)</th>
<th>Ndata/data field (2) (bits)</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>16</td>
<td>144</td>
<td>0</td>
<td>3 &amp; 3</td>
<td>132</td>
<td>126</td>
<td>66</td>
<td>60</td>
</tr>
<tr>
<td>21</td>
<td>16</td>
<td>144</td>
<td>6</td>
<td>3 &amp; 3</td>
<td>132</td>
<td>123</td>
<td>63</td>
<td>60</td>
</tr>
<tr>
<td>22</td>
<td>16</td>
<td>144</td>
<td>12</td>
<td>3 &amp; 3</td>
<td>132</td>
<td>120</td>
<td>63</td>
<td>57</td>
</tr>
<tr>
<td>23</td>
<td>16</td>
<td>144</td>
<td>24</td>
<td>3 &amp; 3</td>
<td>132</td>
<td>114</td>
<td>60</td>
<td>54</td>
</tr>
<tr>
<td>24</td>
<td>16</td>
<td>144</td>
<td>48</td>
<td>3 &amp; 3</td>
<td>132</td>
<td>102</td>
<td>54</td>
<td>48</td>
</tr>
</tbody>
</table>
References

**TDSCDMA_DPCH_DataMux**

**Description**  
Multiplexer for data, TFCI, SS and TPC in DPCH

**Library**  
TDSCDMA, Modems

**Class**  
SDFTDSCDMA_DPCH_DataMux

**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>Link</td>
<td>link selection: Downlink, Uplink</td>
<td></td>
<td>enum</td>
<td></td>
</tr>
<tr>
<td>SpreadFactor</td>
<td>spreading factor</td>
<td>16</td>
<td>int</td>
<td>{1, 2, 4, 8, 16}</td>
</tr>
<tr>
<td>ModType</td>
<td>type of modulation: QPSK, _8PSK</td>
<td></td>
<td>enum</td>
<td></td>
</tr>
<tr>
<td>N_TFCI</td>
<td>number of TFCI bits</td>
<td>0</td>
<td>int</td>
<td></td>
</tr>
<tr>
<td>N_SS_N_TPC</td>
<td>number of SS and TPC</td>
<td>0</td>
<td>int</td>
<td></td>
</tr>
</tbody>
</table>

Values for N_TFCI and N_SS_N_TPC are given in Note 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 of dedicated physical channel</td>
<td>int</td>
</tr>
<tr>
<td>2</td>
<td>TFCI</td>
<td>transport format combination indicator bits</td>
<td>int</td>
</tr>
<tr>
<td>3</td>
<td>SS</td>
<td>information bits for uplink synchronization control</td>
<td>int</td>
</tr>
<tr>
<td>4</td>
<td>TPC</td>
<td>transmit power control 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>5</td>
<td>Output</td>
<td>data other than midamble in DPCH</td>
<td>int</td>
</tr>
</tbody>
</table>
Notes/Equations

1. This model is used to multiplex data, TFCI bits, SS bits and TPC bits for DPCH. Each firing, Bits/slot Output tokens are produced when N\textsubscript{Data/Slot} Data tokens, N\textsubscript{TFCI} TFCI tokens, N\textsubscript{SS} SS tokens, and N\textsubscript{TPC} TPC tokens are consumed.

2. The burst structure is illustrated in Figure 4-6, where time slot n (n = 0 to 6) are the nth traffic time slots, 864-chip duration; DwPTS is downlink pilot time slot, 96-chip duration; UpPTS is uplink pilot time slot, 160-chip duration; GP is main guard period for TDD operation, 96-chip duration.

3. Time slot formats for the downlink with QPSK modulation are given in Table 4-4; time slot formats for uplink with QPSK modulation are given in Table 4-5; time slot formats for both links with 8PSK modulation are given in Table 4-6.

<table>
<thead>
<tr>
<th>Slot Format #</th>
<th>Spread Factor</th>
<th>Midamble Length (chips)</th>
<th>TFCI Code Word (bits)</th>
<th>NSS &amp; TPC (bits)</th>
<th>Bits/Slot</th>
<th>NData/Slot (bits)</th>
<th>NData/Data Field (1) (bits)</th>
<th>NData/Data Field (2) (bits)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>16</td>
<td>144</td>
<td>0</td>
<td>0 &amp; 0</td>
<td>88</td>
<td>88</td>
<td>44</td>
<td>44</td>
</tr>
<tr>
<td>1</td>
<td>16</td>
<td>144</td>
<td>4</td>
<td>0 &amp; 0</td>
<td>88</td>
<td>86</td>
<td>42</td>
<td>44</td>
</tr>
<tr>
<td>2</td>
<td>16</td>
<td>144</td>
<td>8</td>
<td>0 &amp; 0</td>
<td>88</td>
<td>84</td>
<td>42</td>
<td>42</td>
</tr>
<tr>
<td>3</td>
<td>16</td>
<td>144</td>
<td>16</td>
<td>0 &amp; 0</td>
<td>88</td>
<td>80</td>
<td>40</td>
<td>40</td>
</tr>
<tr>
<td>4</td>
<td>16</td>
<td>144</td>
<td>32</td>
<td>0 &amp; 0</td>
<td>88</td>
<td>72</td>
<td>36</td>
<td>36</td>
</tr>
<tr>
<td>5</td>
<td>16</td>
<td>144</td>
<td>0</td>
<td>2 &amp; 2</td>
<td>88</td>
<td>84</td>
<td>44</td>
<td>40</td>
</tr>
<tr>
<td>6</td>
<td>16</td>
<td>144</td>
<td>4</td>
<td>2 &amp; 2</td>
<td>88</td>
<td>82</td>
<td>42</td>
<td>40</td>
</tr>
<tr>
<td>7</td>
<td>16</td>
<td>144</td>
<td>8</td>
<td>2 &amp; 2</td>
<td>88</td>
<td>80</td>
<td>42</td>
<td>38</td>
</tr>
<tr>
<td>8</td>
<td>16</td>
<td>144</td>
<td>16</td>
<td>2 &amp; 2</td>
<td>88</td>
<td>76</td>
<td>40</td>
<td>36</td>
</tr>
<tr>
<td>9</td>
<td>16</td>
<td>144</td>
<td>32</td>
<td>2 &amp; 2</td>
<td>88</td>
<td>68</td>
<td>36</td>
<td>32</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>144</td>
<td>0</td>
<td>0 &amp; 0</td>
<td>1408</td>
<td>1408</td>
<td>704</td>
<td>704</td>
</tr>
</tbody>
</table>

GP = guard period
CP = chip period

Figure 4-6. Burst Structure of Traffic Burst Format
### Table 4-5. Downlink Time Slot Formats

<table>
<thead>
<tr>
<th>Slot Format #</th>
<th>Spread Factor</th>
<th>Midamble Length (chips)</th>
<th>NTFCI Code Word (bits)</th>
<th>NSS &amp; NTPC (bits)</th>
<th>Bits/Slot</th>
<th>NData/Slot (bits)</th>
<th>Ndata/Data Field (1) (bits)</th>
<th>Ndata/Data Field (2) (bits)</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>1</td>
<td>144</td>
<td>4</td>
<td>0 &amp; 0</td>
<td>1408</td>
<td>1406</td>
<td>702</td>
<td>704</td>
</tr>
<tr>
<td>12</td>
<td>1</td>
<td>144</td>
<td>8</td>
<td>0 &amp; 0</td>
<td>1408</td>
<td>1404</td>
<td>702</td>
<td>702</td>
</tr>
<tr>
<td>13</td>
<td>1</td>
<td>144</td>
<td>16</td>
<td>0 &amp; 0</td>
<td>1408</td>
<td>1400</td>
<td>700</td>
<td>700</td>
</tr>
<tr>
<td>14</td>
<td>1</td>
<td>144</td>
<td>32</td>
<td>0 &amp; 0</td>
<td>1408</td>
<td>1392</td>
<td>696</td>
<td>696</td>
</tr>
<tr>
<td>15</td>
<td>1</td>
<td>144</td>
<td>0</td>
<td>2 &amp; 2</td>
<td>1408</td>
<td>1404</td>
<td>704</td>
<td>700</td>
</tr>
<tr>
<td>16</td>
<td>1</td>
<td>144</td>
<td>4</td>
<td>2 &amp; 2</td>
<td>1408</td>
<td>1402</td>
<td>702</td>
<td>700</td>
</tr>
<tr>
<td>17</td>
<td>1</td>
<td>144</td>
<td>8</td>
<td>2 &amp; 2</td>
<td>1408</td>
<td>1400</td>
<td>702</td>
<td>698</td>
</tr>
<tr>
<td>18</td>
<td>1</td>
<td>144</td>
<td>16</td>
<td>2 &amp; 2</td>
<td>1408</td>
<td>1396</td>
<td>700</td>
<td>696</td>
</tr>
<tr>
<td>19</td>
<td>1</td>
<td>144</td>
<td>32</td>
<td>2 &amp; 2</td>
<td>1408</td>
<td>1388</td>
<td>696</td>
<td>692</td>
</tr>
<tr>
<td>20</td>
<td>1</td>
<td>144</td>
<td>0</td>
<td>32 &amp; 32</td>
<td>1408</td>
<td>1344</td>
<td>704</td>
<td>640</td>
</tr>
<tr>
<td>21</td>
<td>1</td>
<td>144</td>
<td>4</td>
<td>32 &amp; 32</td>
<td>1408</td>
<td>1342</td>
<td>702</td>
<td>640</td>
</tr>
<tr>
<td>22</td>
<td>1</td>
<td>144</td>
<td>8</td>
<td>32 &amp; 32</td>
<td>1408</td>
<td>1340</td>
<td>702</td>
<td>638</td>
</tr>
<tr>
<td>23</td>
<td>1</td>
<td>144</td>
<td>16</td>
<td>32 &amp; 32</td>
<td>1408</td>
<td>1336</td>
<td>700</td>
<td>636</td>
</tr>
<tr>
<td>24</td>
<td>1</td>
<td>144</td>
<td>32</td>
<td>32 &amp; 32</td>
<td>1408</td>
<td>1328</td>
<td>696</td>
<td>632</td>
</tr>
</tbody>
</table>

### Table 4-5. Uplink Time Slot Formats

<table>
<thead>
<tr>
<th>Slot Format</th>
<th>Spread Factor</th>
<th>Midamble Length (chips)</th>
<th>NTFCI Code Word (bits)</th>
<th>NSS &amp; NTPC (bits)</th>
<th>Bits/Slot</th>
<th>NData/Slot (bits)</th>
<th>Ndata/Data Field (1) (bits)</th>
<th>Ndata/Data Field (2) (bits)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>16</td>
<td>144</td>
<td>0</td>
<td>0 &amp; 0</td>
<td>88</td>
<td>88</td>
<td>44</td>
<td>44</td>
</tr>
<tr>
<td>1</td>
<td>16</td>
<td>144</td>
<td>4</td>
<td>0 &amp; 0</td>
<td>88</td>
<td>86</td>
<td>42</td>
<td>44</td>
</tr>
<tr>
<td>2</td>
<td>16</td>
<td>144</td>
<td>8</td>
<td>0 &amp; 0</td>
<td>88</td>
<td>84</td>
<td>42</td>
<td>42</td>
</tr>
<tr>
<td>3</td>
<td>16</td>
<td>144</td>
<td>16</td>
<td>0 &amp; 0</td>
<td>88</td>
<td>80</td>
<td>40</td>
<td>40</td>
</tr>
<tr>
<td>4</td>
<td>16</td>
<td>144</td>
<td>32</td>
<td>0 &amp; 0</td>
<td>88</td>
<td>72</td>
<td>36</td>
<td>36</td>
</tr>
<tr>
<td>5</td>
<td>16</td>
<td>144</td>
<td>0</td>
<td>2 &amp; 2</td>
<td>88</td>
<td>84</td>
<td>44</td>
<td>40</td>
</tr>
<tr>
<td>6</td>
<td>16</td>
<td>144</td>
<td>4</td>
<td>2 &amp; 2</td>
<td>88</td>
<td>82</td>
<td>42</td>
<td>40</td>
</tr>
<tr>
<td>7</td>
<td>16</td>
<td>144</td>
<td>8</td>
<td>2 &amp; 2</td>
<td>88</td>
<td>80</td>
<td>42</td>
<td>38</td>
</tr>
<tr>
<td>8</td>
<td>16</td>
<td>144</td>
<td>16</td>
<td>2 &amp; 2</td>
<td>88</td>
<td>76</td>
<td>40</td>
<td>36</td>
</tr>
<tr>
<td>9</td>
<td>16</td>
<td>144</td>
<td>32</td>
<td>2 &amp; 2</td>
<td>88</td>
<td>68</td>
<td>36</td>
<td>32</td>
</tr>
<tr>
<td>10</td>
<td>8</td>
<td>144</td>
<td>0</td>
<td>0 &amp; 0</td>
<td>176</td>
<td>176</td>
<td>88</td>
<td>88</td>
</tr>
<tr>
<td>11</td>
<td>8</td>
<td>144</td>
<td>4</td>
<td>0 &amp; 0</td>
<td>176</td>
<td>174</td>
<td>86</td>
<td>88</td>
</tr>
<tr>
<td>12</td>
<td>8</td>
<td>144</td>
<td>8</td>
<td>0 &amp; 0</td>
<td>176</td>
<td>172</td>
<td>86</td>
<td>86</td>
</tr>
</tbody>
</table>

4-16  TDSCDMA_DPCH_DataMux
<table>
<thead>
<tr>
<th>Slot Format</th>
<th>Spread Factor</th>
<th>Midamble length (chips)</th>
<th>NTFCI code word (bits)</th>
<th>NSS &amp; NTPC (bits)</th>
<th>Bits/ slot</th>
<th>NData/ Slot (bits)</th>
<th>Ndata/ data field (1) (bits)</th>
<th>Ndata/ data field (2) (bits)</th>
</tr>
</thead>
<tbody>
<tr>
<td>13</td>
<td>8</td>
<td>144 16</td>
<td>0 0</td>
<td>176</td>
<td>168</td>
<td>84</td>
<td>84</td>
<td>84</td>
</tr>
<tr>
<td>14</td>
<td>8</td>
<td>144 32</td>
<td>2 &amp; 2</td>
<td>176</td>
<td>160</td>
<td>80</td>
<td>80</td>
<td>80</td>
</tr>
<tr>
<td>15</td>
<td>8</td>
<td>144 16</td>
<td>2 &amp; 2</td>
<td>176</td>
<td>164</td>
<td>84</td>
<td>84</td>
<td>84</td>
</tr>
<tr>
<td>16</td>
<td>8</td>
<td>144 16</td>
<td>2 &amp; 2</td>
<td>176</td>
<td>160</td>
<td>80</td>
<td>80</td>
<td>80</td>
</tr>
<tr>
<td>17</td>
<td>8</td>
<td>144 16</td>
<td>2 &amp; 2</td>
<td>176</td>
<td>164</td>
<td>84</td>
<td>84</td>
<td>84</td>
</tr>
<tr>
<td>18</td>
<td>8</td>
<td>144 16</td>
<td>2 &amp; 2</td>
<td>176</td>
<td>160</td>
<td>80</td>
<td>80</td>
<td>80</td>
</tr>
<tr>
<td>19</td>
<td>8</td>
<td>144 16</td>
<td>2 &amp; 2</td>
<td>176</td>
<td>164</td>
<td>84</td>
<td>84</td>
<td>84</td>
</tr>
<tr>
<td>20</td>
<td>8</td>
<td>144 16</td>
<td>2 &amp; 2</td>
<td>176</td>
<td>160</td>
<td>80</td>
<td>80</td>
<td>80</td>
</tr>
<tr>
<td>21</td>
<td>8</td>
<td>144 16</td>
<td>2 &amp; 2</td>
<td>176</td>
<td>164</td>
<td>84</td>
<td>84</td>
<td>84</td>
</tr>
<tr>
<td>22</td>
<td>8</td>
<td>144 16</td>
<td>2 &amp; 2</td>
<td>176</td>
<td>160</td>
<td>80</td>
<td>80</td>
<td>80</td>
</tr>
<tr>
<td>23</td>
<td>8</td>
<td>144 16</td>
<td>2 &amp; 2</td>
<td>176</td>
<td>164</td>
<td>84</td>
<td>84</td>
<td>84</td>
</tr>
<tr>
<td>24</td>
<td>8</td>
<td>144 16</td>
<td>2 &amp; 2</td>
<td>176</td>
<td>160</td>
<td>80</td>
<td>80</td>
<td>80</td>
</tr>
<tr>
<td>25</td>
<td>4</td>
<td>144 16</td>
<td>2 &amp; 2</td>
<td>176</td>
<td>164</td>
<td>84</td>
<td>84</td>
<td>84</td>
</tr>
<tr>
<td>26</td>
<td>4</td>
<td>144 16</td>
<td>2 &amp; 2</td>
<td>176</td>
<td>160</td>
<td>80</td>
<td>80</td>
<td>80</td>
</tr>
<tr>
<td>27</td>
<td>4</td>
<td>144 16</td>
<td>2 &amp; 2</td>
<td>176</td>
<td>164</td>
<td>84</td>
<td>84</td>
<td>84</td>
</tr>
<tr>
<td>28</td>
<td>4</td>
<td>144 16</td>
<td>2 &amp; 2</td>
<td>176</td>
<td>160</td>
<td>80</td>
<td>80</td>
<td>80</td>
</tr>
<tr>
<td>29</td>
<td>4</td>
<td>144 16</td>
<td>2 &amp; 2</td>
<td>176</td>
<td>164</td>
<td>84</td>
<td>84</td>
<td>84</td>
</tr>
<tr>
<td>30</td>
<td>4</td>
<td>144 16</td>
<td>2 &amp; 2</td>
<td>176</td>
<td>160</td>
<td>80</td>
<td>80</td>
<td>80</td>
</tr>
<tr>
<td>31</td>
<td>4</td>
<td>144 16</td>
<td>2 &amp; 2</td>
<td>176</td>
<td>164</td>
<td>84</td>
<td>84</td>
<td>84</td>
</tr>
<tr>
<td>32</td>
<td>4</td>
<td>144 16</td>
<td>2 &amp; 2</td>
<td>176</td>
<td>160</td>
<td>80</td>
<td>80</td>
<td>80</td>
</tr>
<tr>
<td>33</td>
<td>4</td>
<td>144 16</td>
<td>2 &amp; 2</td>
<td>176</td>
<td>164</td>
<td>84</td>
<td>84</td>
<td>84</td>
</tr>
<tr>
<td>34</td>
<td>4</td>
<td>144 16</td>
<td>2 &amp; 2</td>
<td>176</td>
<td>160</td>
<td>80</td>
<td>80</td>
<td>80</td>
</tr>
<tr>
<td>35</td>
<td>4</td>
<td>144 16</td>
<td>2 &amp; 2</td>
<td>176</td>
<td>164</td>
<td>84</td>
<td>84</td>
<td>84</td>
</tr>
<tr>
<td>36</td>
<td>4</td>
<td>144 16</td>
<td>2 &amp; 2</td>
<td>176</td>
<td>160</td>
<td>80</td>
<td>80</td>
<td>80</td>
</tr>
<tr>
<td>37</td>
<td>4</td>
<td>144 16</td>
<td>2 &amp; 2</td>
<td>176</td>
<td>164</td>
<td>84</td>
<td>84</td>
<td>84</td>
</tr>
<tr>
<td>38</td>
<td>4</td>
<td>144 16</td>
<td>2 &amp; 2</td>
<td>176</td>
<td>160</td>
<td>80</td>
<td>80</td>
<td>80</td>
</tr>
<tr>
<td>39</td>
<td>4</td>
<td>144 16</td>
<td>2 &amp; 2</td>
<td>176</td>
<td>164</td>
<td>84</td>
<td>84</td>
<td>84</td>
</tr>
<tr>
<td>40</td>
<td>2</td>
<td>144 16</td>
<td>2 &amp; 2</td>
<td>176</td>
<td>160</td>
<td>80</td>
<td>80</td>
<td>80</td>
</tr>
<tr>
<td>41</td>
<td>2</td>
<td>144 16</td>
<td>2 &amp; 2</td>
<td>176</td>
<td>164</td>
<td>84</td>
<td>84</td>
<td>84</td>
</tr>
<tr>
<td>42</td>
<td>2</td>
<td>144 16</td>
<td>2 &amp; 2</td>
<td>176</td>
<td>160</td>
<td>80</td>
<td>80</td>
<td>80</td>
</tr>
<tr>
<td>43</td>
<td>2</td>
<td>144 16</td>
<td>2 &amp; 2</td>
<td>176</td>
<td>164</td>
<td>84</td>
<td>84</td>
<td>84</td>
</tr>
<tr>
<td>44</td>
<td>2</td>
<td>144 16</td>
<td>2 &amp; 2</td>
<td>176</td>
<td>160</td>
<td>80</td>
<td>80</td>
<td>80</td>
</tr>
</tbody>
</table>
Table 4-5. Uplink Time Slot Formats

<table>
<thead>
<tr>
<th>Slot Format</th>
<th>Spread</th>
<th>Midamble length (chips)</th>
<th>NTFCI code word (bits)</th>
<th>NSS &amp; NTPC (bits)</th>
<th>Bits/slot</th>
<th>NData/Slot (bits)</th>
<th>Ndata/data field (1) (bits)</th>
<th>Ndata/data field (2) (bits)</th>
</tr>
</thead>
<tbody>
<tr>
<td>45</td>
<td>2</td>
<td>144</td>
<td>0</td>
<td>2 &amp; 2</td>
<td>704</td>
<td>700</td>
<td>352</td>
<td>348</td>
</tr>
<tr>
<td>46</td>
<td>2</td>
<td>144</td>
<td>4</td>
<td>2 &amp; 2</td>
<td>704</td>
<td>698</td>
<td>350</td>
<td>348</td>
</tr>
<tr>
<td>47</td>
<td>2</td>
<td>144</td>
<td>8</td>
<td>2 &amp; 2</td>
<td>704</td>
<td>696</td>
<td>350</td>
<td>346</td>
</tr>
<tr>
<td>48</td>
<td>2</td>
<td>144</td>
<td>16</td>
<td>2 &amp; 2</td>
<td>704</td>
<td>692</td>
<td>348</td>
<td>344</td>
</tr>
<tr>
<td>49</td>
<td>2</td>
<td>144</td>
<td>32</td>
<td>2 &amp; 2</td>
<td>704</td>
<td>684</td>
<td>344</td>
<td>340</td>
</tr>
<tr>
<td>50</td>
<td>2</td>
<td>144</td>
<td>0</td>
<td>16 &amp; 16</td>
<td>704</td>
<td>672</td>
<td>352</td>
<td>320</td>
</tr>
<tr>
<td>51</td>
<td>2</td>
<td>144</td>
<td>4</td>
<td>16 &amp; 16</td>
<td>704</td>
<td>670</td>
<td>350</td>
<td>320</td>
</tr>
<tr>
<td>52</td>
<td>2</td>
<td>144</td>
<td>8</td>
<td>16 &amp; 16</td>
<td>704</td>
<td>668</td>
<td>350</td>
<td>318</td>
</tr>
<tr>
<td>53</td>
<td>2</td>
<td>144</td>
<td>16</td>
<td>16 &amp; 16</td>
<td>704</td>
<td>664</td>
<td>348</td>
<td>316</td>
</tr>
<tr>
<td>54</td>
<td>2</td>
<td>144</td>
<td>32</td>
<td>16 &amp; 16</td>
<td>704</td>
<td>656</td>
<td>344</td>
<td>312</td>
</tr>
<tr>
<td>55</td>
<td>1</td>
<td>144</td>
<td>0</td>
<td>0 &amp; 0</td>
<td>1408</td>
<td>1408</td>
<td>704</td>
<td>704</td>
</tr>
<tr>
<td>56</td>
<td>1</td>
<td>144</td>
<td>4</td>
<td>0 &amp; 0</td>
<td>1408</td>
<td>1406</td>
<td>702</td>
<td>704</td>
</tr>
<tr>
<td>57</td>
<td>1</td>
<td>144</td>
<td>8</td>
<td>0 &amp; 0</td>
<td>1408</td>
<td>1404</td>
<td>702</td>
<td>702</td>
</tr>
<tr>
<td>58</td>
<td>1</td>
<td>144</td>
<td>16</td>
<td>0 &amp; 0</td>
<td>1408</td>
<td>1400</td>
<td>700</td>
<td>700</td>
</tr>
<tr>
<td>59</td>
<td>1</td>
<td>144</td>
<td>32</td>
<td>0 &amp; 0</td>
<td>1408</td>
<td>1392</td>
<td>696</td>
<td>696</td>
</tr>
<tr>
<td>60</td>
<td>1</td>
<td>144</td>
<td>0</td>
<td>2 &amp; 2</td>
<td>1408</td>
<td>1404</td>
<td>704</td>
<td>700</td>
</tr>
<tr>
<td>61</td>
<td>1</td>
<td>144</td>
<td>4</td>
<td>2 &amp; 2</td>
<td>1408</td>
<td>1402</td>
<td>702</td>
<td>700</td>
</tr>
<tr>
<td>62</td>
<td>1</td>
<td>144</td>
<td>8</td>
<td>2 &amp; 2</td>
<td>1408</td>
<td>1400</td>
<td>702</td>
<td>698</td>
</tr>
<tr>
<td>63</td>
<td>1</td>
<td>144</td>
<td>16</td>
<td>2 &amp; 2</td>
<td>1408</td>
<td>1396</td>
<td>700</td>
<td>696</td>
</tr>
<tr>
<td>64</td>
<td>1</td>
<td>144</td>
<td>32</td>
<td>2 &amp; 2</td>
<td>1408</td>
<td>1388</td>
<td>696</td>
<td>692</td>
</tr>
<tr>
<td>65</td>
<td>1</td>
<td>144</td>
<td>0</td>
<td>32 &amp; 32</td>
<td>1408</td>
<td>1344</td>
<td>704</td>
<td>640</td>
</tr>
<tr>
<td>66</td>
<td>1</td>
<td>144</td>
<td>4</td>
<td>32 &amp; 32</td>
<td>1408</td>
<td>1342</td>
<td>702</td>
<td>640</td>
</tr>
<tr>
<td>67</td>
<td>1</td>
<td>144</td>
<td>8</td>
<td>32 &amp; 32</td>
<td>1408</td>
<td>1340</td>
<td>702</td>
<td>638</td>
</tr>
<tr>
<td>68</td>
<td>1</td>
<td>144</td>
<td>16</td>
<td>32 &amp; 32</td>
<td>1408</td>
<td>1336</td>
<td>700</td>
<td>636</td>
</tr>
<tr>
<td>69</td>
<td>1</td>
<td>144</td>
<td>32</td>
<td>32 &amp; 32</td>
<td>1408</td>
<td>1328</td>
<td>696</td>
<td>632</td>
</tr>
</tbody>
</table>
Table 4-6. 8PSK Modulation Time Slot Formats

<table>
<thead>
<tr>
<th>Slot Format</th>
<th>Spread Factor</th>
<th>Midamble Length (chips)</th>
<th>NTFCI Code Word (bits)</th>
<th>NSS &amp; NTPC (bits)</th>
<th>Bits/Slot</th>
<th>NData/Slot (bits)</th>
<th>Ndata/data field (1) (bits)</th>
<th>Ndata/data field (2) (bits)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>144</td>
<td>0</td>
<td>0 &amp; 0</td>
<td>2112</td>
<td>2112</td>
<td>1056</td>
<td>1056</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>144</td>
<td>6</td>
<td>0 &amp; 0</td>
<td>2112</td>
<td>2109</td>
<td>1053</td>
<td>1056</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>144</td>
<td>12</td>
<td>0 &amp; 0</td>
<td>2112</td>
<td>2106</td>
<td>1053</td>
<td>1053</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>144</td>
<td>24</td>
<td>0 &amp; 0</td>
<td>2112</td>
<td>2100</td>
<td>1050</td>
<td>1050</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>144</td>
<td>48</td>
<td>0 &amp; 0</td>
<td>2112</td>
<td>2088</td>
<td>1044</td>
<td>1044</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>144</td>
<td>0</td>
<td>3 &amp; 3</td>
<td>2112</td>
<td>2106</td>
<td>1056</td>
<td>1050</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>144</td>
<td>6</td>
<td>3 &amp; 3</td>
<td>2112</td>
<td>2103</td>
<td>1053</td>
<td>1050</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>144</td>
<td>12</td>
<td>3 &amp; 3</td>
<td>2112</td>
<td>2100</td>
<td>1053</td>
<td>1047</td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>144</td>
<td>24</td>
<td>3 &amp; 3</td>
<td>2112</td>
<td>2094</td>
<td>1050</td>
<td>1044</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>144</td>
<td>48</td>
<td>3 &amp; 3</td>
<td>2112</td>
<td>2082</td>
<td>1044</td>
<td>1038</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>144</td>
<td>0</td>
<td>48 &amp; 48</td>
<td>2112</td>
<td>2016</td>
<td>1056</td>
<td>960</td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>144</td>
<td>6</td>
<td>48 &amp; 48</td>
<td>2112</td>
<td>2013</td>
<td>1053</td>
<td>960</td>
</tr>
<tr>
<td>12</td>
<td>1</td>
<td>144</td>
<td>12</td>
<td>48 &amp; 48</td>
<td>2112</td>
<td>2010</td>
<td>1053</td>
<td>957</td>
</tr>
<tr>
<td>13</td>
<td>1</td>
<td>144</td>
<td>24</td>
<td>48 &amp; 48</td>
<td>2112</td>
<td>2004</td>
<td>1050</td>
<td>954</td>
</tr>
<tr>
<td>14</td>
<td>1</td>
<td>144</td>
<td>48</td>
<td>48 &amp; 48</td>
<td>2112</td>
<td>1992</td>
<td>1044</td>
<td>948</td>
</tr>
<tr>
<td>15</td>
<td>16</td>
<td>144</td>
<td>0</td>
<td>0 &amp; 0</td>
<td>132</td>
<td>132</td>
<td>66</td>
<td>66</td>
</tr>
<tr>
<td>16</td>
<td>16</td>
<td>144</td>
<td>6</td>
<td>0 &amp; 0</td>
<td>132</td>
<td>129</td>
<td>63</td>
<td>66</td>
</tr>
<tr>
<td>17</td>
<td>16</td>
<td>144</td>
<td>12</td>
<td>0 &amp; 0</td>
<td>132</td>
<td>126</td>
<td>63</td>
<td>63</td>
</tr>
<tr>
<td>18</td>
<td>16</td>
<td>144</td>
<td>24</td>
<td>0 &amp; 0</td>
<td>132</td>
<td>120</td>
<td>60</td>
<td>60</td>
</tr>
<tr>
<td>19</td>
<td>16</td>
<td>144</td>
<td>48</td>
<td>0 &amp; 0</td>
<td>132</td>
<td>108</td>
<td>54</td>
<td>54</td>
</tr>
<tr>
<td>20</td>
<td>16</td>
<td>144</td>
<td>0</td>
<td>3 &amp; 3</td>
<td>132</td>
<td>126</td>
<td>66</td>
<td>60</td>
</tr>
<tr>
<td>21</td>
<td>16</td>
<td>144</td>
<td>6</td>
<td>3 &amp; 3</td>
<td>132</td>
<td>123</td>
<td>63</td>
<td>60</td>
</tr>
<tr>
<td>22</td>
<td>16</td>
<td>144</td>
<td>12</td>
<td>3 &amp; 3</td>
<td>132</td>
<td>120</td>
<td>63</td>
<td>57</td>
</tr>
<tr>
<td>23</td>
<td>16</td>
<td>144</td>
<td>24</td>
<td>3 &amp; 3</td>
<td>132</td>
<td>114</td>
<td>60</td>
<td>54</td>
</tr>
<tr>
<td>24</td>
<td>16</td>
<td>144</td>
<td>48</td>
<td>3 &amp; 3</td>
<td>132</td>
<td>102</td>
<td>54</td>
<td>48</td>
</tr>
</tbody>
</table>

References

TDSCDMA_DPCH_Mux

Description  
DPCH multiplexer

Library  
TDSCDMA, Modems

Class  
SDFTDSCDMA_DPCH_Mux

Derived From  
TDSCDMA_CCTrCH_MuxBase

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>PhyChNum_SA</td>
<td>physical channel allocation configuration</td>
<td>00100000</td>
<td>int array</td>
<td>[0, 2] for Uplink, [0, 16] for Downlink</td>
</tr>
<tr>
<td>Link</td>
<td>link selection: Downlink, Uplink</td>
<td>Uplink</td>
<td>enum</td>
<td></td>
</tr>
<tr>
<td>ModType_SA</td>
<td>type of modulation of all slots, 0 for QPSK, 1 for 8PSK</td>
<td>00000000</td>
<td>int array</td>
<td>[0, 1] for each element</td>
</tr>
<tr>
<td>SpreadCode_PA</td>
<td>index of OVSF code corresponding to allocated physical channels</td>
<td>1</td>
<td>int array</td>
<td>[1, SpreadFactor]</td>
</tr>
<tr>
<td>BasicMidambleID</td>
<td>index of basic midamble</td>
<td>1</td>
<td>int</td>
<td>[0, 127]</td>
</tr>
<tr>
<td>K_SA</td>
<td>maximum number of midamble shifts in a cell for all slots</td>
<td>16 16 16 16 16 16 16</td>
<td>int array</td>
<td>[2, 4, 6, 8, 10, 12, 14, 16]</td>
</tr>
<tr>
<td>MidambleID_SA</td>
<td>index of midamble for all slots</td>
<td>5 5 5 5 5 5 5 5</td>
<td>int array</td>
<td>[1, K]</td>
</tr>
<tr>
<td>Gain_PA</td>
<td>gain setting array corresponding to allocated physical channels</td>
<td>1.0</td>
<td>real 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>TFCI</td>
<td>encoded TFCI bits input</td>
<td>int</td>
</tr>
<tr>
<td>2</td>
<td>SS</td>
<td>information bits for uplink synchronization control</td>
<td>int</td>
</tr>
</tbody>
</table>
This model generates signals for several dedicated physical channels (DPCH). The number of DPCHs can be determined by the size of multiple input DataIn dynamically in run time.

The TDSCDMA_DPCH subnetwork generates a signal for one DPCH (refer to the schematic in Figure 6-1 to see its structure). However, the TDSCDMA_DPCH_Mux model is more flexible and can cover all DPCH combinations from flexible rate matching algorithms in the transport channel.

Each firing, 6400 DataOut tokens are produced when 2112 DataIn tokens, 1 SlotFormat token, 48 TFCI, SS and TPC tokens consumed. These are the maximum number of tokens necessary in all cases; the real values needed may be less, padding tokens are filled before this model when needed.

Data of all physical channels are fed from the multiple DataIn pin while TFCI bits, SS and TPC bits are fed in TFCI, SS and TPC pins. The valid size of Data, TFCI, SS and TPC for each DPCH can be calculated from the corresponding input of SlotFormat pin.

2. PhyChNum_SA determines which slots will transmit data and how many physical channels are transferred in one specified slot. It contains 7 elements that represent 7 individual slots. The maximum allocated physical channel number is equal to the sum of the PhyChNum_SA elements. The size of SpreadCode_PA and Gain_PA, the port number of SlotFormat and DataIn must be equal to the sum of the PhyChNum_SA elements.

3. ModType_SA determines the modulation mapping scheme of the data bits: 0 for QPSK, 1 for 8PSK.

---

**Notes/Equations**

- Tabular representation of pin inputs and 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>TPC</td>
<td>transmit power control bits</td>
<td>int</td>
</tr>
<tr>
<td>4</td>
<td>DataIn</td>
<td>bits data stream input before mapping, spreading and scrambling</td>
<td>multiple int</td>
</tr>
<tr>
<td>5</td>
<td>SlotFormat</td>
<td>slot format input corresponding to each physical channel</td>
<td>multiple 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>6</td>
<td>DataOut</td>
<td>output data</td>
<td>complex</td>
</tr>
</tbody>
</table>
4. After modulation, data is spread with corresponding spreading codes. The spreading factors of physical channels are determined by the input of SlotFormat, while the spreading codes index is set by SpreadCode_PA.

5. The index of scramble code is the same as BasicMidambleID.

6. The midamble of each physical channel is determined by UE_Specific based on K_SA and MidambleID_SA settings.

7. Gain_PA determines the gain of each physical channel.

References


TDSCDMA_Demodulator

Description  QPSK, 8PSK demodulator
Library  TDSCDMA, Modems
Class  SDFTDSCDMA_Demodulator

Required Licenses

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Default</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>ModType</td>
<td>type of modulation: QPSK, 8PSK</td>
<td>QPSK</td>
<td>enum</td>
</tr>
<tr>
<td>Decision</td>
<td>decision method of Viterbi or Turbo decoder: Soft decision, Hard decision</td>
<td>Soft decision</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 data from 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>output decision values</td>
<td>real</td>
</tr>
</tbody>
</table>

Notes/Equations

1. This component is used to perform demodulation for QPSK, 8PSK and provide hard- or soft-decision values for Viterbi decoder or Turbo decoder.

Each firing, 2 Output tokens for QPSK, 3 Output tokens for 8PSK are produced when 1 Input token is consumed.

References
Modems

TDSCDMA_Midamble

Description  Midamble generation
Library  TDSCDMA, Modems
Class  SDFTDSCDMA_Midamble
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>MidambleAllocScheme</td>
<td>midamble allocation scheme: UE_Specific, Common, Default</td>
<td>Default</td>
<td>enum</td>
<td></td>
</tr>
<tr>
<td>BasicMidambleID</td>
<td>index of basic midamble</td>
<td>0</td>
<td>int</td>
<td>[0, 127]</td>
</tr>
<tr>
<td>K</td>
<td>maximum number of midamble shifts in a cell</td>
<td>16</td>
<td>int</td>
<td>{2, 4, 6, 8, 10, 12, 14, 16}</td>
</tr>
<tr>
<td>MidambleID</td>
<td>index of midamble</td>
<td>1</td>
<td>int</td>
<td>[1, K]</td>
</tr>
<tr>
<td>SpreadFactor</td>
<td>spreading factor</td>
<td>16</td>
<td>int</td>
<td>{1, 2, 4, 8, 16}</td>
</tr>
<tr>
<td>SpreadCode</td>
<td>index of spread code</td>
<td>1</td>
<td>int</td>
<td>[1, SpreadFactor]</td>
</tr>
<tr>
<td>PhyChNum</td>
<td>number of channelization codes used in a timeslot</td>
<td>1</td>
<td>int</td>
<td>[1, 16]</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>output</td>
<td>midamble output</td>
<td>complex</td>
</tr>
</tbody>
</table>

Notes/Equations

1. This model is used to generate midamble sequence used in TD-SCDMA. Each firing, one token is produced.
2. Midambles of different users active in the same cell and the same time slot are cyclically shifted versions of one basic midamble code. The value of BasicMidambleID determines the index of the basic midamble to be used. The applicable basic midambles are given in Annex B.1 of [1], 128 totally. The basic midamble codes in Annex B.1 are listed in hexadecimal notation. The binary form is derived as given in Table 4-7.
3. For each particular basic midamble code, its binary representation can be written as
\[ m_p = (m_1, m_2, \ldots, m_p) \]
where \( P = 128 \). As QPSK modulation is used, the midamble is transformed into a complex form \( m_p = (m_1, m_2, \ldots, m_p) \). The relation between \( m_p \) and \( m_p \) is given by:
\[ m_i = (j)^i m_i \] where \( i = 1, \ldots, P \)
Hence, the elements \( m_i \) of \( m_p \) are alternating real and imaginary.
To derive the required midamble, \( m_p \) is periodically extended to the size
\[ i_{\text{max}} = L_m + (K-1) W, \]
where

<table>
<thead>
<tr>
<th>Binary Elements ( m_i )</th>
<th>Hexadecimal Digit Mapping</th>
</tr>
</thead>
<tbody>
<tr>
<td>-1 -1 -1 -1</td>
<td>0</td>
</tr>
<tr>
<td>-1 -1 -1 +1</td>
<td>1</td>
</tr>
<tr>
<td>-1 -1 +1 -1</td>
<td>2</td>
</tr>
<tr>
<td>-1 -1 +1 +1</td>
<td>3</td>
</tr>
<tr>
<td>-1 +1 -1 -1</td>
<td>4</td>
</tr>
<tr>
<td>-1 +1 -1 +1</td>
<td>5</td>
</tr>
<tr>
<td>-1 +1 +1 -1</td>
<td>6</td>
</tr>
<tr>
<td>-1 +1 +1 +1</td>
<td>7</td>
</tr>
<tr>
<td>+1 -1 -1 -1</td>
<td>8</td>
</tr>
<tr>
<td>+1 -1 -1 +1</td>
<td>9</td>
</tr>
<tr>
<td>+1 -1 +1 -1</td>
<td>A</td>
</tr>
<tr>
<td>+1 -1 +1 +1</td>
<td>B</td>
</tr>
<tr>
<td>+1 +1 -1 -1</td>
<td>C</td>
</tr>
<tr>
<td>+1 +1 -1 +1</td>
<td>D</td>
</tr>
<tr>
<td>+1 +1 +1 -1</td>
<td>E</td>
</tr>
<tr>
<td>+1 +1 +1 +1</td>
<td>F</td>
</tr>
</tbody>
</table>
L_m = 144, is the midamble length
K = 2, 4, 6, 8, 10, 12, 14, 16, is the maximum number of different midamble shifts in a cell
W = \lceil \frac{P}{K} \rceil, is the shift between midambles and \lfloor x \rfloor denotes the largest number less or equal to x.
P = 128, is the length of basic midamble.

So a new vector \( \mathbf{m} \) is obtained
\[ \mathbf{m} = (m_1, m_2, \ldots, m_{i_{max}}) \]
The first P elements of \( \mathbf{m} \) are the same as those in \( \mathbf{m}_p \), the following elements repeat the beginning:
\[ \mathbf{m} = \mathbf{m}_{i_p} \text{ for the subset } i = (P + 1), \ldots, i_{\text{max}} \]
The midamble for user k, \( \mathbf{m}^{(k)} \) of length L_m is derived using \( \mathbf{m} \), which can be written as
\[ \mathbf{m}^{(k)} = (m_1^{(k)}, m_2^{(k)}, \ldots, m_{L_m}^{(k)}) \]
The L_m midamble elements \( \mathbf{m}_i^{(k)} \) are generated for each midamble of the k users (k=1, ..., K) based on
\[ \mathbf{m}_i^{(k)} = \mathbf{m}_{i + (K - k)W} \text{ with } i = 1, \ldots, L_m \text{ and } k = 1, \ldots, K \]
The derived midambles have complex values and are not subject to channelization or scrambling.
4. There are three midamble allocation schemes.

- **UE specific midamble allocation**: a UE specific midamble for DL and UL is explicitly assigned by higher layers.

- **Default midamble allocation**: the midamble for DL and UL is assigned by layer 1 depending on associated channelization code.

- **Common midamble allocation**: the midamble for DL is allocated by layer 1 depending on the number of channelization codes currently present in the DL time slot.

In the implementation of this model:

- if `MidambleAllocScheme=UE_Specific`, only the `BasicMidambleID`, `K`, and `MidambleID` parameters are used to specify which midamble is exported, the values of the other parameters are ignored.

- if `MidambleAllocScheme=Common`, only the `BasicMidambleID`, `K`, and `PhyChNum` parameters are used to specify which midamble is exported, the values of the other parameters are ignored.

- if `MidambleAllocScheme=Default`, only the `BasicMidambleID`, `K`, `SpreadFactor`, and `SpreadCode` parameters are used to specify which midamble is exported, the values of the other parameters are ignored.

**References**

TDSCDMA_Modulator

Description  Modulator to generate QPSK and 8-PSK modulation symbols
Library  TDSCDMA, Modems
Class  SDFTDSCDMA_Modulator
Required Licenses

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Default</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>ModType</td>
<td>type of modulation: QPSK, 8PSK</td>
<td>QPSK</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 data</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>output modulation symbols</td>
<td>complex</td>
</tr>
</tbody>
</table>

Notes/Equations

1. This model is used to map the bits from the output of the physical channel mapping onto the signal point constellation for QPSK and 8PSK modulation. Each firing,
   • for QPSK, 1 output token is produced when 2 input tokens are consumed.
   • for 8PSK, 1 output token is produced when 3 input tokens are consumed.
2. QPSK data modulation is performed to the bits from the output of the physical channel mapping and combines 2 consecutive binary bits to a complex valued data symbol. Each user burst has two data carrying parts, termed data blocks:

\[ \mathbf{d}^{(k,i)} = \left( \mathbf{d}_1^{(k,i)}, \mathbf{d}_2^{(k,i)}, \ldots, \mathbf{d}_{N_k}^{(k,i)} \right), \quad i = 1, 2; k = 1, \ldots, K_{\text{Code}} \]

\( K_{\text{Code}} \) is the number of codes used in a time slot, max \( K_{\text{Code}} = 6 \). \( N_k \) is the number of symbols per data field for the code \( k \). This number is linked to the spreading factor.

Data symbols \( \mathbf{d}^{(k,i)} \) are generated from two consecutive data bits from the output of the physical channel mapping procedure

\[ b_{l,n}^{(k,i)} \in \{0, 1\} \quad l = 1, 2; \quad k = 1, \ldots, K_{\text{Code}}; \quad n = 1, \ldots, N_k; i = 1, 2 \]

using Table 4-8.

<table>
<thead>
<tr>
<th>Input (consecutive binary bit pattern)</th>
<th>Output (complex symbol)</th>
</tr>
</thead>
<tbody>
<tr>
<td>( b_{1,n}^{(k,i)} b_{2,n}^{(k,i)} )</td>
<td>( \mathbf{d}^{(k,i)} )</td>
</tr>
<tr>
<td>00</td>
<td>+1</td>
</tr>
<tr>
<td>01</td>
<td>-1</td>
</tr>
<tr>
<td>10</td>
<td>-1</td>
</tr>
<tr>
<td>11</td>
<td>-j</td>
</tr>
</tbody>
</table>

3. 8PSK data modulation is performed to the bits from the output of the physical channel mapping procedure; 3 consecutive binary bits are represented by one complex valued data symbol. Each user burst has two data carrying parts, termed data blocks:

\[ \mathbf{d}^{(k,i)} = \left( \mathbf{d}_1^{(k,i)}, \mathbf{d}_2^{(k,i)}, \ldots, \mathbf{d}_{N_k}^{(k,i)} \right), \quad i = 1, 2; k = 1, \ldots, K_{\text{Code}} \]

\( K_{\text{Code}} \) is the number of codes used in a time slot, max \( K_{\text{Code}} = 6 \). \( N_k \) is the number of symbols per data field for the code \( k \). This number is linked to the spreading factor.
Data symbols $d^{(k,i)}$ are generated from 3 consecutive data bits from the output of the physical channel mapping procedure:

$$b_{l,n}^{(k,i)} \in \{0, 1\} \quad l = 1, 2, 3; \quad k = 1, \ldots, K_{\text{Code}}; \quad n = 1, \ldots$$

using Table 4-9.

<table>
<thead>
<tr>
<th>Input (Consecutive binary bit pattern)</th>
<th>Output (complex symbol)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$b_{0,n}^{(k,i)}b_{2,n}^{(k,i)}b_{1,n}^{(k,i)}$</td>
<td>$d^{(k,i)}$</td>
</tr>
<tr>
<td>000</td>
<td>$\cos(11\pi/8) + j\sin(11\pi/8)$</td>
</tr>
<tr>
<td>001</td>
<td>$\cos(9\pi/8) + j\sin(9\pi/8)$</td>
</tr>
<tr>
<td>010</td>
<td>$\cos(5\pi/8) + j\sin(5\pi/8)$</td>
</tr>
<tr>
<td>011</td>
<td>$\cos(7\pi/8) + j\sin(7\pi/8)$</td>
</tr>
<tr>
<td>100</td>
<td>$\cos(13\pi/8) + j\sin(13\pi/8)$</td>
</tr>
<tr>
<td>101</td>
<td>$\cos(15\pi/8) + j\sin(15\pi/8)$</td>
</tr>
<tr>
<td>110</td>
<td>$\cos(3\pi/8) + j\sin(3\pi/8)$</td>
</tr>
<tr>
<td>111</td>
<td>$\cos(\pi/8) + j\sin(\pi/8)$</td>
</tr>
</tbody>
</table>

References

TDSCDMA_OnePhyCh

Description  One physical channel
Library    TDSCDMA, Modems
Class     SDFTDSCDMA_OnePhyCh
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>SlotIndex</td>
<td>index of slot</td>
<td>2</td>
<td>int</td>
<td>[0, 6]</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>Burst</td>
<td>burst</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>SubFrm</td>
<td>subframe with only one burst</td>
<td>complex</td>
</tr>
</tbody>
</table>

Notes/Equations

1. This model forms one physical channel using a burst.
   Each firing, 6400 SubFrm tokens are produced when 864 Burst tokens are consumed.

2. To simplify combining of the physical channels, each physical channel is placed at a specific interval in one subframe according to the SlotIndex parameter setting. Special models implement DwPTS and UpPTS.

3. The sub-frame structure is illustrated in Figure 4-7. Where Time slot #n (n from 0 to 6) are the nth traffic time slot, 864 chips duration; DwPTS is the
downlink pilot time slot, 96 chips duration; UpPTS is the uplink pilot time slot, 160 chips duration; GP is the main guard period for TDD operation, 96 chips duration. The total number of traffic time slots for uplink and downlink is 7, and each traffic time slot is 864 chips duration.

Among the 7 traffic time slots, time slot 0 is always allocated as downlink while time slot 1 is always allocated as uplink. Uplink and downlink time slots are separated by switching points. Between downlink and uplink time slots, the special period is the switching point to separate uplink and downlink. In each sub-frame of 5ms for 1.28Mcps option, there are two switching points (uplink to downlink and vice versa).

![Figure 4-7. Structure of Subframe for 1.28Mcps TDD Option](image-url)

**References**

TDSCDMA_OnePhyChDeMux

Description  One physical channel demultiplexer
Library   TDSCDMA, Modems
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>SlotIndex</td>
<td>index of slot</td>
<td>2</td>
<td>int</td>
<td>[0, 6]</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>SubFrm</td>
<td>input subframe</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>slot</td>
<td>output time slot</td>
<td>complex</td>
</tr>
</tbody>
</table>

Notes/Equations

1. This subnetwork is used to demultiplex a specified burst from one physical channel.
   
   The schematic for this subnetwork is shown in Figure 4-8.

   Each firing, 6400 tokens are consumed when 864 tokens produced.
2. The sub-frame structure is illustrated in Figure 4-9. The slot is chopped and output according to specified slot index.

References

TDSCDMA_OVSF

Description  OVSF code generation
Library  TDSCDMA, Modems
Class  SDFTDSCDMA_OVSF
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>SpreadFactor</td>
<td>spreading factor</td>
<td>16</td>
<td>int</td>
<td>{1, 2, 4, 8, 16}</td>
</tr>
<tr>
<td>SpreadCode</td>
<td>index of OVSF code</td>
<td>1</td>
<td>int</td>
<td>[1, SpreadFactor]</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>output</td>
<td>OVSF code output</td>
<td>complex</td>
</tr>
</tbody>
</table>

Notes/Equations

1. This model is used to generate OVSF codes used in TD-SCDMA. Each firing, one token is produced.
2. The OVSF codes are derived from the code tree. Each code at each level with length \( l \) will generate two codes at the next level with length \( 2 \times l \). The first \( l \) elements of the two son codes are the same as the \( l \) elements of the father code, and the last \( l \) elements of the son code with lower index are also the same as the \( l \) elements of the father code, whereas, the last \( l \) elements of the son code with higher index are opposite to the \( l \) elements of the father code.

Associated with each OVSF code is a multiplier.
Modems

\[
\begin{align*}
\omega_{Q_k}^{(k)}
\end{align*}
\]

taking values from the set

\[
\left\{ e^{\frac{j\pi}{Q_k}p_k} \right\}
\]

where \( p_k \) is a permutation of the integer set \( \{0, \ldots, Q_k-1\} \) and \( Q_k \) the spreading factor.

Values of the multiplier of each channelization code are given in Table 4-10.

The output of this model is the product of the specified OVSF code and its corresponding multiplier.

<table>
<thead>
<tr>
<th>( k )</th>
<th>( \omega_{Q = 1}^{(k)} )</th>
<th>( \omega_{Q = 2}^{(k)} )</th>
<th>( \omega_{Q = 4}^{(k)} )</th>
<th>( \omega_{Q = 8}^{(k)} )</th>
<th>( \omega_{Q = 16}^{(k)} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>-j</td>
<td>1</td>
<td>-1</td>
</tr>
<tr>
<td>2</td>
<td>-j</td>
<td>1</td>
<td>+j</td>
<td>-j</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>+j</td>
<td>+j</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>-1</td>
<td>-1</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>-j</td>
<td>+j</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>-1</td>
<td>-1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>-j</td>
<td>-1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>-j</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>+j</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>+j</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>-j</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>-j</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>+j</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>-1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

References

4-38  TDSCDMA_OVSF
Modems

TDSCDMA_PSCH_DataMux

Description  Time division multiplexer for physical uplink/downlink data
Library  TDSCDMA, Modems
Class  SDFTDSCDMA_PSCH_DataMux
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>Link</td>
<td>link selection: Downlink, Uplink</td>
<td>Downlink</td>
<td>enum</td>
<td></td>
</tr>
<tr>
<td>SpreadFactor</td>
<td>spreading factor</td>
<td>16</td>
<td>int</td>
<td>{1, 16} for downlink; (1,2,4,8,16) for uplink</td>
</tr>
<tr>
<td>ModType</td>
<td>type of modulation: QPSK, 8PSK</td>
<td>QPSK</td>
<td>enum</td>
<td></td>
</tr>
<tr>
<td>N_TFCI</td>
<td>number of TFCI bits</td>
<td>0</td>
<td>int</td>
<td>{0, 4,8,16,32} for QPSK; {0,6,12,24,48} for 8PSK</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 of dedicated physical channel</td>
<td>int</td>
</tr>
<tr>
<td>2</td>
<td>TFCI</td>
<td>transport format combination indicator 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>3</td>
<td>Output</td>
<td>data other than midamble in PSCH</td>
<td>int</td>
</tr>
</tbody>
</table>

Notes/Equations

4-40  TDSCDMA_PSCH_DataMux
1. This model is used to multiplex data, TFCI bits for PD/USCH.

Each firing,

• for QPSK, 1408/SpreadFactor Output tokens are produced when
  (1408/SpreadFactor-N_TFCI/2) Data tokens and N_TFCI/2 TFCI tokens are
  consumed.

• for 8PSK, 2112/SpreadFactor Output tokens are produced when
  (2112/SpreadFactor-N_TFCI/2) Data tokens and N_TFCI/2 TFCI tokens are
  consumed.

Physical downlink/uplink shared channel provides TFCI transmission.

References
Group Radio Access Network; Physical channels and mapping of transport
channels onto physical channels (TDD) (Release 4), version 4.3.0, Dec., 2001
Modems

TDSCDMA_Scramble

Description: Scramble generation
Library: TDSCDMA, Modems
Class: SDFTTDSCDMA_Scramble

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>ScrambleCode</td>
<td>index of scramble code</td>
<td>0</td>
<td>int</td>
<td>[0, 127]</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>output</td>
<td>scramble code output</td>
<td>complex</td>
</tr>
</tbody>
</table>

Notes/Equations

1. This model is used to generate the scramble code used in TD-SCDMA. Each firing, one token is produced.

2. Spreading of data consists of channelization and scrambling operations.

Each complex valued data symbol is spread with a real channelization code of length $Q_k \in \{1, 2, 4, 8, 16\}$. The resulting sequence is then scrambled by a cell specific complex scrambling sequence $\gamma$ of length 16,

where

$$\gamma = (\gamma_1, \gamma_2, ..., \gamma_{16})^*$$

The complex scrambling code $\gamma$ is generated from the binary scrambling code

$$\nu = (\nu_1, \nu_2, ..., \nu_{16})$$
The available binary scrambling codes are given in Annex A of [1], 128 totally. The relation between the elements of $\gamma$ and $\nu$ is given by:

$$\gamma = (j)^i \nu_i.$$  

where $\nu_i \in \{1, -1\}, i = 1, \ldots, 16$

Hence, the elements $\nu_i$ of $\gamma$ are alternating real and imaginary.

References

TDSCDMA_Sync

Description  Sychronization code generation
Library  TDSCDMA, Modems
Class  SDFTDSCDMA_Sync
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>LinkDir</td>
<td>link direction: Down, Up</td>
<td>Down</td>
<td>enum</td>
<td></td>
</tr>
<tr>
<td>SyncCode</td>
<td>index of basic synchronization code</td>
<td>0</td>
<td>int</td>
<td>([0, 31]) when LinkDir=Down; ([0, 255]) when LinkDir=Up</td>
</tr>
<tr>
<td>ModPhase</td>
<td>type of modulation quadruples, valid only in downlink: S1, S2</td>
<td>S1</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>output</td>
<td>sync code output</td>
<td>complex</td>
</tr>
</tbody>
</table>

Notes/Equations

1. This model generates the SYNC_DL and SYNC_UL sequences. Each firing, one token is produced.

2. DwPTS is composed of 64 chips of a complex SYNC_DL sequence \(s = (s_1, s_2, ..., s_{64})\) and 32 chips of guard period. SYNC_DL code is not scrambled. To generate the complex SYNC_DL code, the basic SYNC_DL code \(s = (s_1, s_2, ..., s_{64})\) is used. There are 32 different basic SYNC_DL codes for the whole system. The relation between \(s\) and \(\bar{s}\) is given by:
Hence, the elements $s_i$ of $s$ are alternating real and imaginary.

3. The SYNC_DL is QPSK modulated; the SYNC_DL phase is used to signal the presence of the P-CCPCH in the multi-frame of the resource units of the first two code channels in time slot 0.

The SYNC_DL sequences are modulated with respect to the midamble ($m^{(1)}$) in time slot 0. Four consecutive phases (phase quadruple) of the SYNC_DL are used to indicate the presence of the P-CCPCH in the following 4 sub-frames. When the presence of a P-CCPCH is indicated, the following sub-frame is the first subframe of the interleaving period. As QPSK is used for the modulation of the SYNC-DL, the phase 45, 135, 225 and 315 are used.

The total number of different phase quadruples is 2 (S1 and S2). A quadruple always starts with an even system frame number ((SFN mod 2)=0). Table 4-11 describes the quadruples.

<table>
<thead>
<tr>
<th>Name</th>
<th>Phase Quadruple</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1</td>
<td>135,45,225,135</td>
<td>There is a P-CCPCH in the next 4 sub-frames</td>
</tr>
<tr>
<td>S2</td>
<td>315,225,315,45</td>
<td>There is no P-CCPCH in the next 4 sub-frames</td>
</tr>
</tbody>
</table>

In the implementation of this model,

- if LinkDir=Down and ModPhase=S1, SYNC_DL sequences $s = (s_1, s_2, ..., s_{64})$ in every 4 subframes are rotated additionally with angles of 135, 45, 225, and 135 degrees.
- if LinkDir=Down and ModPhase=S2, SYNC_DL sequences $s = (s_1, s_2, ..., s_{64})$ in every 4 subframes are rotated additionally with angles of 315, 225, 315, and 45 degrees.
- If LinkDir=Up, the model ignores the value of ModPhase and the SYNC_DL sequence $s$ does not have additional rotation.

4. UpPTS is composed of 128 chips of a complex SYNC_UL sequence $s = (s_1, s_2, ..., s_{128})$ and 32 chips of guard period.
The SYNC_UL code is not scrambled.

For UL code, the basic SYNC_DL code \( \xi = (s_1, s_2, \ldots, s_{128}) \) is used.

There are 256 different basic SYNC_UL codes for the whole system. The relation between \( \xi \) and \( s \) is given by:

\[
\xi_i = (j)^i s_i \text{ where } v_i \in \{1, -1\}, i = 1, \ldots, 128
\]

Hence, the elements \( s_i \) of \( \xi \) are alternating real and imaginary.

References

Chapter 5: Multiplexing and Coding Components
Multiplexing and Coding Components

**TDSCDMA_1stDeIntlvr**

**Description**  First deinterleaver
**Library**  TDSCDMA, Multiplexing & Coding
**Class**  SDFTDSCDMA_1stDeIntlvr
**Derived From**  TDSCDMA_ChDecodingBase
**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>DynTF_Set</td>
<td>dynamic part of Transport Format Set</td>
<td>244 488</td>
<td>int array</td>
<td>†</td>
</tr>
<tr>
<td>TTI</td>
<td>transmission time interval: TTI_10ms, TTI_20ms, TTI_40ms, TTI_80ms</td>
<td>TTI_10ms</td>
<td>enum</td>
<td></td>
</tr>
<tr>
<td>CRC</td>
<td>length of CRC bits: No_CRC, CRC_8_bits, CRC_12_bits, CRC_16_bits, CRC_24_bits</td>
<td>CRC_16_bits</td>
<td>enum</td>
<td></td>
</tr>
<tr>
<td>ChCodingType</td>
<td>channel coding type: No_Coding, CC_HalfRate, CC_OneThirdRate, TurboCoding</td>
<td>CC_HalfRate</td>
<td>enum</td>
<td></td>
</tr>
</tbody>
</table>

† The array structure of DynTF_Set is [transport block size 1, transport block set size 1, transport block size 2, transport block set size 2, ...].

The value range of transport block size is [0, 5000].
The value range of transport block set size is [0, 20000].
Transport block set size must be an integer multiple of transport block size.

### 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>DataI</td>
<td>input data</td>
<td>real</td>
</tr>
<tr>
<td>2</td>
<td>TFI_I</td>
<td>transport format indicator</td>
<td>int</td>
</tr>
</tbody>
</table>
This model implements reverse process of first interleaver.

Each firing, 1 TFI_O token and N DataO tokens are produced when 1 TFI_I and N DataI tokens consumed, while N is calculated using the maximum value of transport block set size, that is, the maximum valid data in one TTI.

TFI value is an index used to select the transport block size and transport block set size from the transport format set, as specified by DynTF_Set. The minimal TFI is 0, the step is 1. The number of input and output tokens in each firing is calculated using the maximum value of transport block set size, while the valid tokens is calculated using the transport block set indexed by TFI.

The first interleaving is a block interleaver with inter-column permutations. This model recover the order of data in one TTI.

References

Multiplexing and Coding Components

**TDSCDMA_1stIntlvr**

Description  First interleaver  
Library  TDSCDMA, Multiplexing & Coding  
Class  SDFTDSCDMA_1stIntlvr  
Derived From  TDSCDMA_ChEncodingBase  
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>DynTF_Set</td>
<td>dynamic part of Transport Format Set</td>
<td>244 488</td>
<td>int array</td>
<td>†</td>
</tr>
<tr>
<td>TTI</td>
<td>transmission time interval: TTI_10ms, TTI_20ms, TTI_40ms, TTI_80ms</td>
<td>TTI_10ms</td>
<td>enum</td>
<td></td>
</tr>
<tr>
<td>CRC</td>
<td>length of CRC bits: No_CRC, CRC_8_bits, CRC_12_bits, CRC_16_bits, CRC_24_bits</td>
<td>CRC_16_bits</td>
<td>enum</td>
<td></td>
</tr>
<tr>
<td>ChCodingType</td>
<td>channel coding type: No_Coding, CC_HalfRate, CC_OneThirdRate, TurboCoding</td>
<td>CC_HalfRate</td>
<td>enum</td>
<td></td>
</tr>
</tbody>
</table>

† The array structure of DynTF_Set is \([\text{transport block size } 1, \text{transport block set size } 1, \text{transport block size } 2, \text{transport block set size } 2, \ldots]\).

The value range of transport block size is \([0, 5000]\).
The value range of transport block set size is \([0, 20000]\).
Transport block set size must be an integer multiple of transport block size.

### 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>DataI</td>
<td>transport block set</td>
<td>int</td>
</tr>
<tr>
<td>2</td>
<td>TFI_I</td>
<td>transport format indicator</td>
<td>int</td>
</tr>
</tbody>
</table>

5-4  TDSCDMA_1stIntlvr
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>DataO</td>
<td>transport block set with CRC attached</td>
<td>int</td>
</tr>
<tr>
<td>4</td>
<td>TFI_O</td>
<td>transport format indicator</td>
<td>int</td>
</tr>
</tbody>
</table>

Notes/Equations

1. This model implements block interleaving.
2. Each firing, 1 TFI_O token and N DataO tokens are produced when 1 TFI_I and N DataI tokens consumed, while N is calculated using the maximum value of transport block set size, that is, the maximum valid data in one TTI.
3. TFI value is an index used to select the transport block size and transport block set size from the transport format set, as specified by DynTF_Set. The minimal TFI is 0, the step is 1. The number of input and output tokens in each firing is calculated using the maximum value of transport block set size, while the valid tokens is calculated using the transport block set indexed by TFI.
4. The first interleaving is a block interleaver with inter-column permutations.

References

Multiplexing and Coding Components

TDSCDMA_2ndDeIntlvr

Description  Second deinterleaver
Library  TDSCDMA, Multiplexing & Coding
Class  SDFTDSCDMA_2ndDeIntlvr
Derived From  TDSCDMA_CCTrCH_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>PhyChNunm_SA</td>
<td>physical channel allocation</td>
<td>0 0 1 0 0 0 0</td>
<td>int array</td>
<td>[0, 2] for Uplink, [0, 16] for Downlink</td>
</tr>
<tr>
<td>IntlvrMethod</td>
<td>interleaving method for the second interleaver: Frame, Slot</td>
<td>Slot</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>DataIn</td>
<td>input data</td>
<td>multiple real</td>
</tr>
<tr>
<td>2</td>
<td>SizeInM</td>
<td>input data length</td>
<td>multiple 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>DataOut</td>
<td>output data</td>
<td>multiple real</td>
</tr>
<tr>
<td>4</td>
<td>SizeOutM</td>
<td>output data length</td>
<td>multiple int</td>
</tr>
</tbody>
</table>

Notes/Equations

1. This model performs the inverse operation of the second interleaving. Each firing, this model consumes $704 \times 3 \times 2$ interleaved tokens for each physical channel.
channel on multiple pin DataIn, which is the maximum number of data bits one physical channel can contain in one frame. The tokens consist of valid ones and padding ones. 1 token for each physical channel is consumed on multiple pin SizeInM to indicate the number of valid tokens on DataIn. \(704 \times 3 \times 2\) deinterleaved tokens are exported for each physical channel on multiple pin DataOut, which also consists of valid ones and padding ones. 1 token for each physical channel is consumed on multiple pin SizeOutM to indicate the number of valid tokens on DataOut.

2. PhyChNum_SA indicates the number of physical channels allocated in each time slot.

3. Second interleaving can be applied jointly to all data bits transmitted during one frame, or separately within each time slot, on which the CCTrCH is mapped. IntlvrMethod indicates which method is used.

4. For details regarding second interleaving, refer to [1].

References

TDSCDMA_2ndIntlvr

Description  Second interleaver
Library  TDSCDMA, Multiplexing & Coding
Class  SDFTDSCDMA_2ndIntlvr
Derived From  TDSCDMA_CCTrCH_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>PhyChNum_SA</td>
<td>physical channel allocation configuration</td>
<td>00100000</td>
<td>int array</td>
<td>[0, 2] for Uplink, [0, 16] for Downlink</td>
</tr>
<tr>
<td>IntlvrMethod</td>
<td>interleaving method for the second interleaver: Frame, Slot</td>
<td>Slot</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>DataIn</td>
<td>input data</td>
<td>multiple int</td>
</tr>
<tr>
<td>2</td>
<td>SizeInM</td>
<td>input data length</td>
<td>multiple 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>DataOut</td>
<td>output data</td>
<td>multiple int</td>
</tr>
<tr>
<td>4</td>
<td>SizeOutM</td>
<td>output data length</td>
<td>multiple int</td>
</tr>
</tbody>
</table>

Notes/Equations

1. This model performs second interleaving, which acts as a block interleaver and consists of bits input to a matrix with padding, the inter-column permutation
for the matrix and bits output from the matrix with pruning. Each firing, this model consumes $704 \times 3 \times 2$ tokens for each physical channel on multiple pin DataIn, which is the maximum number of data bits one physical channel can contain in one frame. The tokens consist of valid ones and padding ones. 1 token for each physical channel is consumed on multiple pin SizeInM to indicate the number of valid tokens on DataIn. $704 \times 3 \times 2$ interleaved tokens are exported for each physical channel on multiple pin DataOut, which also consists of valid ones and padding ones. 1 token for each physical channel is consumed on multiple pin SizeOutM to indicate the number of valid tokens on DataOut.

2. PhyChNum_SA indicates the number of physical channels allocated in each time slot.

3. Second interleaving can be applied jointly to all data bits transmitted during one frame, or separately within each time slot, on which the CCTrCH is mapped. IntlvrMethod indicates which method is used.

4. For details regarding second interleaving, refer to [1].

References

TDSCDMA_BitScrambling

Description  
Bit scrambling

Library  
TDSCDMA, Multiplexing & Coding

Class  
SDFTDSCDMA_BitScrambling

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>PhyChNum_SA</td>
<td>physical channel allocation</td>
<td>0 0 1 0 0 0</td>
<td>int array</td>
<td>[0, 16]</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>DataIn</td>
<td>input data</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>DataOut</td>
<td>output data</td>
<td>int</td>
</tr>
</tbody>
</table>

Notes/Equations

1. This model implements bit scrambling.

2. Each firing, PhyChNumAll × MAX_BIT_SLOT tokens are consumed at DataIn, and PhyChNumAll × MAX_BIT_SLOT tokens are exported at DataOut. PhyChNumAll is the number of allocated physical channels and MAX_BIT_SLOT is the maximum number of bits possible in one physical channel, that is 704 × 3 × 2.
3. The bits output from the transport channel multiplexer are scrambled by bit scrambler. The input bits to the bit scrambler are denoted by \( h_1, h_2, h_3, \ldots, h_s \), where \( S \) is the number of bits input to the bit scrambling block equal to the total number of bits on the CCTrCH. The bits after bit scrambling are denoted by \( s_1, s_2, s_3, \ldots, s_s \). Bit scrambling is defined by the following action:

\[
S_k = h_k \oplus p_k
\]

where \( k = 1, 2, \ldots, S \) and \( p_k \) results from the following operation:

\[
p_k = \left( \sum_{i=1}^{16} g_i \times p_{k-1} \right) \mod 2; p_k = 0; (k < 1); p_1 = 1; g = \{0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 1\}
\]

References

Multiplexing and Coding Components

TDSCDMA_CRC_Decoder

Description  CRC decoder for transport block
Library  TDSCDMA, Multiplexing & Coding
Class  SDFTDSCDMA_CRC_Decoder
Derived From  TDSCDMA_ChCodingBase
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>DynTF_Set</td>
<td>dynamic part of Transport Format Set</td>
<td>244 488</td>
<td>int array</td>
<td>†</td>
</tr>
<tr>
<td>CRC</td>
<td>length of CRC bits: No_CRC, CRC_8_bits, CRC_12_bits, CRC_16_bits, CRC_24_bits</td>
<td>CRC_16_bits</td>
<td>enum</td>
<td></td>
</tr>
</tbody>
</table>

† The array structure of DynTF_Set is [transport block size 1, transport block set size 1, transport block size 2, transport block set size 2, ...]

The value range of transport block size is [0, 5000].
The value range of transport block set size is [0, 20000].
Transport block set size must be an integer multiple of transport block size.

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>DataI</td>
<td>transport block set</td>
<td>int</td>
</tr>
<tr>
<td>2</td>
<td>TFI_I</td>
<td>transport format indicator</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>DataO</td>
<td>transport block set with CRC attached</td>
<td>int</td>
</tr>
<tr>
<td>4</td>
<td>error</td>
<td>packet error indicator</td>
<td>int</td>
</tr>
</tbody>
</table>
Notes/Equations

1. This model adds CRC bits to each transport block.

2. Each firing, 1 error token and N DataO tokens are produced when 1 TFI_I and (N+CRC \times m) Data tokens consumed, while N is the maximum value of transport block set size, m is N divided by corresponding transport block size and CRC is the length of CRC bits.

3. TFI value is an index used to select the transport block size and transport block set size from the transport format set, as specified by DynTF_Set. The minimal TFI is 0, the step is 1. The number of input and output tokens in each firing is calculated using the maximum value of transport block set size, while the valid tokens is calculated using the transport block set indexed by TFI.

4. CRC bits are attached to each transport block.

The entire transport block is used to calculate the CRC parity bits for each transport block. The parity bits are generated by one of the following cyclic generator polynomials:

\begin{align*}
g_{CRC24}(D) &= D^{24} + D^{23} + D^6 + D^5 + D + 1 \\
g_{CRC16}(D) &= D^{16} + D^{12} + D^5 + 1 \\
g_{CRC12}(D) &= D^{12} + D^{11} + D^3 + D^2 + D + 1 \\
g_{CRC8}(D) &= D^8 + D^7 + D^4 + D^3 + D + 1
\end{align*}

If transport blocks are not input to the CRC calculation \((M_i = 0)\), a CRC will not be attached; if transport blocks are input to the CRC calculation \((M_i \neq 0)\) and the size of a transport block is zero \((A_i = 0)\), a CRC will be attached (all parity bits equal to zero).

The bits after the CRC attachment are denoted by \(b_{m1}, b_{m2}, b_{m3}, \ldots, b_{mB_i}\), where \(B_i = A_i + L_i\). The relation between \(a_{mk}\) and \(b_{mk}\) is:

\begin{align*}
b_{mk} &= a_{mk} \\
k &= 1, 2, 3, \ldots, A_i \\
b_{mk} &= p_{m}(L+1-(K-A_i)), \\
k &= A_i + 1, A_i + 2, A_i + 3, \ldots, A_i + L_i
\end{align*}
5. The model regenerates the CRC bits and compares with the received CRC bits for each transport block. If any are different, the transport block will be marked as a wrong block. The number of wrong blocks of each firing is the output of error.

References

TDSCDMA_CRC_Encoder

Description  CRC generator for transport block
Library  TDSCDMA, Multiplexing & Coding
Class  SDFTDSCDMA_CRC_Encoder
Derived From  TDSCDMA_ChEncodingBase
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>DynTF_Set</td>
<td>dynamic part of Transport Format Set</td>
<td>244 488</td>
<td>int array</td>
<td>†</td>
</tr>
<tr>
<td>CRC</td>
<td>length of CRC bits: No_CRC, CRC_8_bits, CRC_12_bits, CRC_16_bits, CRC_24_bits</td>
<td>CRC_16_bits</td>
<td>enum</td>
<td></td>
</tr>
</tbody>
</table>

† The array structure of DynTF_Set is [transport block size 1, transport block set size 1, transport block size 2, transport block set size 2, ...]
The value range of transport block size is [0, 5000].
The value range of transport block set size is [0, 20000].
Transport block set size must be an integer multiple of transport block size.

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>DataI</td>
<td>transport block set</td>
<td>int</td>
</tr>
<tr>
<td>2</td>
<td>TFI_I</td>
<td>transport format indicator</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>DataO</td>
<td>transport block set with CRC attached</td>
<td>int</td>
</tr>
<tr>
<td>4</td>
<td>TFI_O</td>
<td>transport format indicator</td>
<td>int</td>
</tr>
</tbody>
</table>
Notes/Equations

1. This model adds CRC bits to each transport block.

2. Each firing, 1 TFI_O token and \((N + \text{CRC} \times m)\) DataO tokens are produced when 1 TFI_I and N DataI tokens consumed, while N is the maximum value of transport block set size, m is N divided by corresponding transport block size and CRC is the length of CRC bits.

3. TFI value is an index used to select the transport block size and transport block set size from the transport format set, as specified by DynTF_Set. The minimal TFI is 0, the step is 1. The number of input and output tokens in each firing is calculated using the maximum value of transport block set size, while the valid tokens is calculated using the transport block set indexed by TFI.

4. CRC bits are attached to each transport block.

The entire transport block is used to calculate the CRC parity bits for each transport block. The parity bits are generated by one of the following cyclic generator polynomials:

\[ g_{\text{CRC24}}(D) = D^{24} + D^{23} + D^6 + D^5 + D + 1 \]
\[ g_{\text{CRC16}}(D) = D^{16} + D^{12} + D^5 + 1 \]
\[ g_{\text{CRC12}}(D) = D^{12} + D^{11} + D^3 + D^2 + D + 1 \]
\[ g_{\text{CRC8}}(D) = D^8 + D^7 + D^4 + D^3 + D + 1 \]

If no transport blocks are input to the CRC calculation \((M_i = 0)\), no CRC attachment will be done. If transport blocks are input to the CRC calculation \((M_i \neq 0)\) and the size of a transport block is zero \( (A_i = 0)\), CRC must be attached, i.e. all parity bits equal to zero.

The bits after CRC attachment are denoted by \(b_{im1}, b_{im2}, b_{im3}, ..., b_{imB} \), where \(B_i = A_i + L_i\). The relation between \(a_{imk}\) and \(b_{imk}\) is:

\[ b_{imk} = a_{imk} \]
\[ k = 1, 2, 3, ..., A_i \]

\[ b_{imk} = p_{im}(L + 1 - (K - A_i)), \]
\[ k = A_i + 1, A_i + 2, A_i + 3, ..., A_i + L_i \]

References
Multiplexing and Coding Components

**TDSCDMA_ChCoding**

**Description**  
Channel coding

**Library**  
TDSCDMA, Multiplexing & Coding

**Class**  
SDFTDSCDMA_ChCoding

**Derived From**  
TDSCDMA_ChEncodingBase

**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>DynTF_Set</td>
<td>dynamic part of Transport Format Set</td>
<td>244 488</td>
<td>int array</td>
<td>†</td>
</tr>
<tr>
<td>CRC</td>
<td>length of CRC bits: No_CRC, CRC_8_bits,</td>
<td>CRC_16_bits</td>
<td>enum</td>
<td></td>
</tr>
<tr>
<td></td>
<td>CRC_12_bits, CRC_16_bits, CRC_24_bits</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ChCodingType</td>
<td>channel coding type: No_Coding, CC_HalfRate,</td>
<td>CC_HalfRate</td>
<td>enum</td>
<td></td>
</tr>
<tr>
<td></td>
<td>CC_OneThirdRate, TurboCoding</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

† The array structure of DynTF_Set is [transport block size 1, transport block set size 1, transport block size 2, transport block set size 2, ...].  
The value range of transport block size is [0, 5000].  
The value range of transport block set size is [0, 20000].  
Transport block set size must be an integer multiple of transport block size.

**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>Data1</td>
<td>transport block set</td>
<td>int</td>
</tr>
<tr>
<td>2</td>
<td>TFI_I</td>
<td>transport format indicator</td>
<td>int</td>
</tr>
</tbody>
</table>
Notes/Equations

1. This model implements channel coding.

2. Each firing, 1 TFI_O token and N DataO tokens are produced when 1 TFI_I and M DataI tokens consumed, while N and M are calculated using the maximum value of transport block set size, that is, the maximum valid data in one TTI after and before channel coding.

TFI value is an index used to select the transport block size and transport block set size from the transport format set, as specified by DynTF_Set. The minimal TFI is 0, the step is 1. The number of input and output tokens in each firing is calculated using the maximum value of transport block set size, while the valid tokens is calculated using the transport block set indexed by TFI.

3. The following channel coding schemes can be applied to transport channels:
   - convolutional coding
   - turbo coding
   - no coding

Usage of coding scheme and coding rate for the different types of TrCH is given in Table 5-1. The values of $Y_i$ in connection with each coding scheme:

- convolutional coding with rate 1/2: $Y_i = 2 \times K_i + 16$; rate 1/3: $Y_i = 3 \times K_i + 24$;
- turbo coding with rate 1/3: $Y_i = 3 \times K_i + 12$;
- no coding: $Y_i = K_i$. 

---

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>DataO</td>
<td>transport block set with CRC attached</td>
<td>int</td>
</tr>
<tr>
<td>4</td>
<td>TFI_O</td>
<td>transport format indicator</td>
<td>int</td>
</tr>
</tbody>
</table>
where $Y_i$ is the number of encoded bits, and $K_i$ is the number of bits in each code block.

Table 5-1. Channel Coding Schemes and Rates for 1.28 Mcps TDD

<table>
<thead>
<tr>
<th>Type of TrCH</th>
<th>Coding scheme</th>
<th>Coding rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>BCH</td>
<td>Convolutional coding</td>
<td>1/3</td>
</tr>
<tr>
<td>PCH</td>
<td>1/3, 1/2</td>
<td></td>
</tr>
<tr>
<td>RACH</td>
<td>1/2</td>
<td></td>
</tr>
<tr>
<td>DCH, DSCH, FACH, USCH</td>
<td>1/3, 1/2</td>
<td></td>
</tr>
<tr>
<td>Turbo coding</td>
<td>1/3</td>
<td></td>
</tr>
<tr>
<td>No coding</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

4. Convolutional codes with constraint length 9 and coding rates 1/3 ($G_0=557$ (octal), $G_1=663$ (octal), $G_2=711$ (octal)) and 1/2 ($G_0=561$ (octal), $G_1=753$ (octal)) are defined.

5. The scheme of Turbo coder is a parallel concatenated convolutional code (PCCC) with two 8-state constituent encoders and one Turbo code internal interleaver. The coding rate of Turbo coder is 1/3.

The transfer function of the 8-state constituent code for PCCC is:

$$G(D) = [1, g_1(D)/g_0(D)]$$

where

$$g_0(D) = 1 + D^2 + D^3$$
$$g_1(D) = 1 + D + D^3$$

Trellis termination is performed by taking the tail bits from the shift register feedback after all information bits are encoded. Tail bits are padded after the encoding of information bits.

References

TDSCDMA_ChDecoding

Description  Channel decoding
Library  TDSCDMA, Multiplexing & Coding
Class  SDFTDSCDMA_ChDecoding
Derived From  TDSCDMA_ChCodingBase
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>DynTF_Set</td>
<td>dynamic part of Transport Format Set</td>
<td>244 488</td>
<td>int array</td>
<td>†</td>
</tr>
<tr>
<td>CRC</td>
<td>length of CRC bits: No_CRC, CRC_8_bits, CRC_12_bits, CRC_16_bits, CRC_24_bits</td>
<td>CRC_16_bits</td>
<td>enum</td>
<td></td>
</tr>
<tr>
<td>ChCodingType</td>
<td>channel coding type: No_Coding, CC_HalfRate, CC_OneThirdRate, TurboCoding</td>
<td>CC_HalfRate</td>
<td>enum</td>
<td></td>
</tr>
<tr>
<td>TC_Iterative</td>
<td>times of iterative decoding in turbo decoder</td>
<td>4</td>
<td>int</td>
<td>[1, 10]</td>
</tr>
</tbody>
</table>

† The array structure of DynTF_Set is [transport block size 1, transport block set size 1, transport block size 2, transport block set size 2, ...]
The value range of transport block size is [0, 5000].The value range of transport block set size is [0, 20000].Transport block set size must be an integer multiple of transport block size.

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>DataI</td>
<td>input data</td>
<td>real</td>
</tr>
<tr>
<td>2</td>
<td>TFI_I</td>
<td>transport format indicator</td>
<td>int</td>
</tr>
</tbody>
</table>
Multiplexing and Coding 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>DataO</td>
<td>output data</td>
<td>int</td>
</tr>
<tr>
<td>4</td>
<td>TFI_O</td>
<td>transport format indicator</td>
<td>int</td>
</tr>
</tbody>
</table>

Notes/Equations

1. This model implements channel decoding.

2. Each firing, 1 TFI_O token and N DataO tokens are produced when 1 TFI_I and M DataI tokens are consumed, while M and N are calculated using the maximum value of transport block set size, that is, the maximum valid data in one TTI before and after channel coding.

   TFI value is an index used to select the transport block size and transport block set size from the transport format set, as specified by DynTF_Set. The minimal TFI is 0, the step is 1. The number of input and output tokens in each firing is calculated using the maximum value of transport block set size, while the valid tokens is calculated using the transport block set indexed by TFI.

3. These channel coding schemes can be applied to transport channels:
   - convolutional coding;
   - turbo coding;
   - no coding.

Usage of coding scheme and coding rate for the different types of TrCH is given in Table 5-2. The values of $Y_i$ in connection with each coding scheme:

   - convolutional coding with rate 1/2: $Y_i = 2 \times K_i + 16$; rate 1/3: $Y_i = 3 \times K_i + 24$;
   - turbo coding with rate 1/3: $Y_i = 3 \times K_i + 12$;
   - no coding: $Y_i = K_i$.

where $Y_i$ is the number of encoded bits, and $K_i$ is the number of bits in each code block.
4. Convolutional codes with constraint length 9 and coding rates 1/3 \((G_0=557 \text{ (octal)}, G_1=663 \text{ (octal)}, G_2=711 \text{ (octal)})\) and 1/2 \((G_0=561 \text{ (octal)}, G_1=753 \text{ (octal)})\) are defined.

5. This model uses viterbi algorithm to decode convolutional code.

6. The scheme of turbo coder is a parallel concatenated convolutional code (PCCC) with two 8-state constituent encoders and one Turbo code internal interleaver. The coding rate of turbo coder is 1/3.

   The transfer function of the 8-state constituent code for PCCC is:
   \[
   G(D) = [1, g_1(D)/g_0(D)],
   \]
   where
   \[
   g_0(D) = 1 + D^2 + D^3,
   \]
   \[
   g_1(D) = 1 + D + D^3.
   \]

   Trellis termination is performed by taking the tail bits from the shift register feedback after all information bits are encoded. Tail bits are padded after the encoding of information bits.

7. This model performs turbo code decoding with MAP algorithm (Maximum A Posterior). It is a modified BCJR algorithm for RSC code. Two parallel concatenated MAP decoders constitute the turbo code decoder.

References


Multiplexing and Coding Components


TDSCDMA_CodeBlkSeg

Description  Code block segmentation
Library  TDSCDMA, Multiplexing & Coding
Class  SDFTDSCDMA_CodeBlkSeg
Derived From  TDSCDMA_ChEncodingBase
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>DynTF_Set</td>
<td>dynamic part of Transport Format Set</td>
<td>244 488</td>
<td>int array</td>
<td>†</td>
</tr>
<tr>
<td>CRC</td>
<td>length of CRC bits: No_CRC, CRC_8_bits, CRC_12_bits, CRC_16_bits, CRC_24_bits</td>
<td>CRC_16_bits</td>
<td>enum</td>
<td></td>
</tr>
<tr>
<td>ChCodingType</td>
<td>channel coding type: No_Coding, CC_HalfRate, CC_OneThirdRate, TurboCoding</td>
<td>CC_HalfRate</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>DataI</td>
<td>transport block set</td>
<td>int</td>
</tr>
<tr>
<td>2</td>
<td>TFI_I</td>
<td>transport format indicator</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>DataO</td>
<td>transport block set with CRC attached</td>
<td>int</td>
</tr>
<tr>
<td>4</td>
<td>TFI_O</td>
<td>transport format indicator</td>
<td>int</td>
</tr>
</tbody>
</table>
Notes/Equations

1. This model implements transport block concatenation and code block segmentation.

2. Each firing, 1 TFI_I token and N DataO tokens are produced when 1 TFI_I and M DataI tokens consumed, while N and M is calculated using the maximum value of transport block set size, that is, the maximum valid data in one TTI after and before adding possible filler bits.

3. TFI value is an index used to select the transport block size and transport block set size from the transport format set, as specified by DynTF_Set. The minimal TFI is 0, the step is 1. The number of input and output tokens in each firing is calculated using the maximum value of transport block set size, while the valid tokens is calculated using the transport block set indexed by TFI.

4. All transport blocks in a TTI are serially concatenated. If the number of bits in a TTI is larger than the maximum size of a code block, then code block segmentation is performed after the concatenation of the transport blocks. The maximum size of the code blocks depends on whether convolutional, turbo coding or no coding is used for the TrCH.

Segmentation of the bit sequence from transport block concatenation is performed if \( X_i \geq Z \). The code blocks after segmentation are of the same size. The number of code blocks on TrCH \( i \) is denoted by \( C_i \). If the number of bits input to the segmentation, \( X_i \), is not a multiple of \( C_i \), filler bits are added to the beginning of the first block. If turbo coding is selected and \( X_i < 40 \), filler bits are added to the beginning of the code block. The filler bits are transmitted and they are always set to 0. The maximum code block sizes are:

- convolutional coding: \( Z = 504 \);
- turbo coding: \( Z = 5114 \);
- no channel coding: \( Z = \) unlimited.

References

TDSCDMA_DeCodeBlkSeg

Description  Code block desegmentation
Library  TDSCDMA, Multiplexing & Coding
Class  SDFTDSCDMA_DeCodeBlkSeg
Derived From  TDSCDMA_ChEncodingBase
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>DynTF_Set</td>
<td>dynamic part of Transport Format Set</td>
<td>244 488</td>
<td>int array</td>
<td>†</td>
</tr>
<tr>
<td>CRC</td>
<td>length of CRC bits: No_CRC, CRC_8_bits, CRC_12_bits, CRC_16_bits, CRC_24_bits</td>
<td>CRC_16_bits</td>
<td>enum</td>
<td></td>
</tr>
<tr>
<td>ChCodingType</td>
<td>channel coding type: No_Coding, CC_HalfRate, CC_OneThirdRate, TurboCoding</td>
<td>CC_HalfRate</td>
<td>enum</td>
<td></td>
</tr>
</tbody>
</table>

† The array structure of DynTF_Set is \([\text{transport block size} \, \text{transport block set size} \, \text{transport block size} \, \text{transport block set size} \, \ldots]\).
The value range of transport block size is \([0, 5000]\).
The value range of transport block set size is \([0, 20000]\).
Transport block set size must be an integer multiple of transport block size.

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>DataI</td>
<td>transport block set</td>
<td>int</td>
</tr>
<tr>
<td>2</td>
<td>TFI_I</td>
<td>transport format indicator</td>
<td>int</td>
</tr>
</tbody>
</table>
Multiplexing and Coding 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>DataO</td>
<td>transport block set with CRC attached</td>
<td>int</td>
</tr>
<tr>
<td>4</td>
<td>TFI_O</td>
<td>transport format indicator</td>
<td>int</td>
</tr>
</tbody>
</table>

Notes/Equations

1. This model implements reverse process of transport block concatenation and code block segmentation.

2. Each firing, 1 TFI_I token and N DataO tokens are produced when 1 TFI_I and M DataI tokens consumed, while N and M is calculated using the maximum value of transport block set size, that is, the maximum valid data in one TTI after and before removing possible filler bits.

3. TFI value is an index used to select the transport block size and transport block set size from the transport format set, as specified by DynTF_Set. The minimal TFI is 0, the step is 1. The number of input and output tokens in each firing is calculated using the maximum value of transport block set size, while the valid tokens is calculated using the transport block set indexed by TFI.

4. All transport blocks in a TTI are serially concatenated. If the number of bits in a TTI is larger than the maximum size of a code block, then code block segmentation is performed after the concatenation of the transport blocks. The maximum size of the code blocks depends on whether convolutional, turbo coding or no coding is used for the TrCH.

Segmentation of the bit sequence from transport block concatenation is performed if \( X_i > Z \). The code blocks after segmentation are of the same size. The number of code blocks on TrCH \( i \) is denoted by \( C_i \). If the number of bits input to the segmentation, \( X_i \), is not a multiple of \( C_i \), filler bits are added to the beginning of the first block. If turbo coding is selected and \( X_i < 40 \), filler bits are added to the beginning of the code block. The filler bits are transmitted and they are always set to 0. The maximum code block sizes are:

- convolutional coding: \( Z = 504 \)
- turbo coding: \( Z = 5114 \)
- no channel coding: \( Z = \text{unlimited} \).

5. This model removes the possible filler bits.
References

Multiplexing and Coding Components

**TDSCDMA_DePhyChMap**

**Description**  
Physical channel demapping

**Library**  
TDSCDMA, Multiplexing & Coding

**Class**  
SDFTDSCDMA_DePhyChMap

**Derived From**  
TDSCDMA_CCTrCH_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>PhyChNum_SA</td>
<td>physical channel allocation configuration</td>
<td>0 0 1 0 0 0 0</td>
<td>int array</td>
<td>[0, 2] for Uplink, [0, 16] for Downlink</td>
</tr>
<tr>
<td>Link</td>
<td>link selection: Downlink, Uplink</td>
<td>Downlink</td>
<td>enum</td>
<td></td>
</tr>
<tr>
<td>ModType_SA</td>
<td>type of modulation of all slots, 0 for QPSK, 1 for 8PSK</td>
<td>0 0 0 0 0 0 0</td>
<td>int array</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>DataI</td>
<td>input data</td>
<td>multiple real</td>
</tr>
<tr>
<td>2</td>
<td>SltFmtIn</td>
<td>input data slot format</td>
<td>multiple 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>DataO</td>
<td>output data</td>
<td>multiple real</td>
</tr>
<tr>
<td>4</td>
<td>SizeOutM</td>
<td>output data length</td>
<td>multiple int</td>
</tr>
</tbody>
</table>

---

**Notes/Equations**

5-30   TDSCDMA_DePhyChMap
1. This model performs the inverse operation of physical channel mapping. Each firing, this model consumes $704 \times 3$ mapped tokens for each physical channel on multiple pin DataI, which is the maximum number of tokens one physical channel can contain in one subframe. 1 token for each physical channel on multiple pin SltFmtIn is consumed to indicate the slot format index for the physical channel. $704 \times 3$ demapped tokens which consist of valid and padding ones are exported for each physical channel in one subframe on multiple pin DataO and 1 token is exported on multiple pin SizeOutM to indicate the number of valid tokens for the physical channel on DataO.

2. PhyChNum_SA indicates the number of physical channels allocated in each time slot.

3. For details regarding physical channel mapping algorithm, refer to [1].

References

Multiplexing and Coding Components

**TDSCDMA_DePhyChSeg**

**Description**  
Physical channel desegmentation

**Library**  
TDSCDMA, Multiplexing & Coding

**Class**  
SDFTDSCDMA_DePhyChSeg

**Derived From**  
TDSCDMA_CCTrCH_MuxBase

**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>PhyChNum_SA</td>
<td>physical channel allocation configuration</td>
<td>0 0 1 0 0 0 0</td>
<td>int array</td>
<td>[0, 2] for Uplink, [0, 16] for Downlink</td>
</tr>
<tr>
<td>TrChNum</td>
<td>number of Transport Channels</td>
<td>2</td>
<td>int</td>
<td>[1, 32]</td>
</tr>
<tr>
<td>RM_TA</td>
<td>rate matching attributes of all Transport Channels</td>
<td>1 1</td>
<td>int array</td>
<td>[1, 256]</td>
</tr>
<tr>
<td>DynTF_Set_TA</td>
<td>dynamic part of TF set of all Transport Channels</td>
<td>100 100 244 244</td>
<td>int array</td>
<td>[0, 5000] for transport block size, [0, 20000] for transport block set size</td>
</tr>
<tr>
<td>TF_SetSize_TA</td>
<td>transport format set size of all Transport Channels</td>
<td>1 1</td>
<td>int array</td>
<td>[1, 64] for each element</td>
</tr>
<tr>
<td>TTI_TA</td>
<td>transmission time interval of all Transport Channels</td>
<td>2 1</td>
<td>int array</td>
<td>[0, 3] for each element</td>
</tr>
<tr>
<td>CRC_TA</td>
<td>number of CRC bits of all Transport Channels</td>
<td>2 3</td>
<td>int array</td>
<td>[0, 4] for each element</td>
</tr>
<tr>
<td>ChCodingType_TA</td>
<td>channel coding type of all Transport Channels</td>
<td>2 2</td>
<td>int array</td>
<td>[0, 3] for each element</td>
</tr>
<tr>
<td>PuncLimit</td>
<td>puncturing limit</td>
<td>2/3</td>
<td>real</td>
<td>(0, 1]</td>
</tr>
<tr>
<td>Link</td>
<td>link selection: Downlink, Uplink</td>
<td></td>
<td>enum</td>
<td></td>
</tr>
<tr>
<td>ModType_SA</td>
<td>type of modulation of all slots, 0 for QPSK, 1 for 8PSK</td>
<td>0 0 0 0 0 0 0</td>
<td>int array</td>
<td>[0, 1] for each element</td>
</tr>
</tbody>
</table>
This model is used to perform physical channel desegmentation, the inverse operation of physical channel segmentation.

Each firing, MAX_BIT_SLOT tokens for each physical channel are consumed for each physical channel at multiple pin DataIn, in which MAX_BIT_SLOT is the possible maximum number of tokens in one physical channel, that is $704 \times 3 \times 2$; 1 token consumed at TFCI indicates the value of transport format combination indicator; bm_PhyChNumAll $\times$ MAX_BIT_SLOT tokens are exported at DataOut, in which bm_PhyChNumAll is the number of allocated physical channels.

Each firing, this model consumes the tokens in all allocated physical channels and combines them into one CCTrCH data block.
Multiplexing and Coding Components

The bits input to the physical channel desegmentation are denoted by $u_{p1}, u_{p2}, u_{p3}, ..., u_{pU_p}$, $p=1,2,...,P$, where $p$ is physical channel number and $U_p$ is the number of bits in physical channel $p$, $P$ is number of physical channels. The output bits are denoted by $x_1, x_2, x_3, ..., x_Y$, where $Y = U_1 + U_2 + ... U_P$. The relation between $x_k$ and $u_{pk}$ is given below.

\[
\begin{align*}
  u_{1,k} &= x_{i,k} & k &= 1, 2, ..., U_1 \\
  u_{2,k} &= x_{i,k+U} & k &= 1, 2, ..., U_2 \\
  u_{P,k} &= x_{i,k+(P-1)U} & k &= 1, 2, ..., U_P
\end{align*}
\]

The $x_1, x_2, x_3, ..., x_Y$ is exported at DataOut, if $Y$ is less than $b_{m_PhyChNumAll} \times MAX\_BIT\_SLOT$, padding bits (0) are added.

3. All transport channel information must be provided in the form of arrays.

For DynTF_Set_TA the correct form is transport block size 1, transport block set size 1, transport block size 2, transport block set size 2, etc. The size of this array must be a multiple of 2, and the transport block set size must be a multiple of the relative transport block size.

When setting TTI_TA, CRC_TA and ChCodingType_TA, refer to Table 5-3.

Table 5-3. Array Values

<table>
<thead>
<tr>
<th>TTI_TA</th>
<th>Value</th>
<th>CRC_TA</th>
<th>Coding</th>
<th>Value</th>
<th>ChCodingType_TA</th>
<th>Coding</th>
</tr>
</thead>
<tbody>
<tr>
<td>10ms</td>
<td>0</td>
<td>No CRC</td>
<td>0</td>
<td>No Coding</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>20ms</td>
<td>1</td>
<td>8 bits</td>
<td>1</td>
<td>1/2 CC</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>40ms</td>
<td>2</td>
<td>12 bits</td>
<td>2</td>
<td>1/3 CC</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>80ms</td>
<td>3</td>
<td>16 bits</td>
<td>3</td>
<td>1/3 TC</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>24 bits</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

CC = convolutional coding; TC = turbo coding

4. PuncLimit denotes the variable PL defined in [2]. Refer to [2] for details regarding use of this variable in rate matching algorithm.

5. PhyChNum_SA indicates the number of allocated physical channels in each slot. The sum of PhyChNum_SA elements is the number of allocated physical channels.

6. TFCI_SA indicates in which slots TFCI bits will be transmitted. 0 denotes no TFCI bits will be transmitted in the slot. 1 denotes TFCI bits can be
transmitted in the slot. The setting must be consistent with PhyChNum_SA setting, which means TFCI bits can only be transmitted in those slots in which the elements of PhyChNum_SA are not zero. Only the first allocated physical channel in each slot is used to transmit TFCI bits.

7. TFCI_Length_SA indicates the number of TFCI bits transmitted in each slot. If 0 is selected, the number of TFCI bits transmitted in the slot is dependent on the value of TFCI imported at TFCI pin. If a non-zero is selected, the number of TFCI bits transmitted in the slot is this non-zero value.

8. MinSF_PA indicates the minimum spread factor that can be used for corresponding physical channel. The size of MinSF_PA must be equal to the sum of PhyChNum_SA elements.

9. NdataOption specifies how the target number of rate-matched data is calculated with MinSF_PA. For Downlink, only Minimum can be selected. For Uplink, both can be selected. Refer to [2] for details.

10. SS_TPC_SA indicates the number of SS and TPC symbols transmitted in each slot. 1 denotes one SS and one TPC symbols are transmitted in the slot; 2 denotes no SS and no TPC symbols are transmitted in the slot; 3 denotes 16/SF SS and 16/SF TPC symbols are transmitted in the slot, where SF is the spreading factor of the physical channel used to transmitted SS and TPC symbols. Only the first allocated physical channel in each slot is used to transmit SS and TPC symbols.

References


TDSCDMA_DeRadioEqual

Description  Radio frame size deequalization
Library    TDSCDMA, Multiplexing & Coding
Class  SDFTDSCDMA_DeRadioEqual
Derived From  TDSCDMA_ChDecodingBase
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>DynTF_Set</td>
<td>dynamic part of Transport Format Set</td>
<td>244 48B</td>
<td>int array</td>
<td>†</td>
</tr>
<tr>
<td>TTI</td>
<td>transmission time interval: TTI_10ms, TTI_20ms, TTI_40ms, TTI_80ms</td>
<td>TTI_10ms</td>
<td>enum</td>
<td></td>
</tr>
<tr>
<td>CRC</td>
<td>length of CRC bits: No_CRC, CRC_8_bits, CRC_12_bits, CRC_16_bits, CRC_24_bits</td>
<td>CRC_16_bits</td>
<td>enum</td>
<td></td>
</tr>
<tr>
<td>ChCodingType</td>
<td>channel coding type: No_Coding, CC_HalfRate, CC_OneThirdRate, TurboCoding</td>
<td>CC_HalfRate</td>
<td>enum</td>
<td></td>
</tr>
</tbody>
</table>

† The array structure of DynTF_Set is [transport block size 1, transport block set size 1, transport block size 2, transport block set size 2, ...].

The value range of transport block size is [0, 5000].
The value range of transport block set size is [0, 20000].
Transport block set size must be an integer multiple of transport block size.

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>DataI</td>
<td>input data</td>
<td>real</td>
</tr>
<tr>
<td>2</td>
<td>TFI_I</td>
<td>transport format indicator</td>
<td>int</td>
</tr>
</tbody>
</table>
1. This model implements reverse process of radio frame size equalization.

2. Each firing, 1 TFI_I token and N DataO tokens are produced when 1 TFI_I and M DataI tokens consumed, while N and M are calculated using the maximum value of transport block set size, that is, the maximum valid data in one TTI after and before radio frame size deequalization.

3. TFI value is an index used to select the transport block size and transport block set size from the transport format set, as specified by DynTF_Set. The minimal TFI is 0, the step is 1. The number of input and output tokens in each firing is calculated using the maximum value of transport block set size, while the valid tokens is calculated using the transport block set indexed by TFI.

4. Radio frame size equalization is padding the input bit sequence in order to ensure that the output can be averaged into radio frames if the number of radio frames in one TTI is larger than 1.

5. Radio frame size deequalization removes the padding bits.

References
Multiplexing and Coding Components

**TDSCDMA_DeRadioSeg**

**Description**  Radio frame desegmentation  
**Library**  TDSCDMA, Multiplexing & Coding  
**Class**  SDFTDSCDMA_DeRadioSeg  
**Derived From**  TDSCDMA_ChDecodingBase  
**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>DynTF_Set</td>
<td>dynamic part of Transport Format Set</td>
<td>244 488</td>
<td>int array</td>
<td>†</td>
</tr>
<tr>
<td>TTI</td>
<td>transmission time interval: TTI_10ms, TTI_20ms, TTI_40ms, TTI_80ms</td>
<td>TTI_10ms</td>
<td>enum</td>
<td></td>
</tr>
<tr>
<td>CRC</td>
<td>length of CRC bits: No_CRC, CRC_8_bits, CRC_12_bits, CRC_16_bits, CRC_24_bits</td>
<td>CRC_16_bits</td>
<td>enum</td>
<td></td>
</tr>
<tr>
<td>ChCodingType</td>
<td>channel coding type: No_Coding, CC_HalfRate, CC_OneThirdRate, TurboCoding</td>
<td>CC_HalfRate</td>
<td>enum</td>
<td></td>
</tr>
</tbody>
</table>

† The array structure of DynTF_Set is [transport block size 1, transport block set size 1, transport block size 2, transport block set size 2, ...].  
The value range of transport block size is [0, 5000].  
The value range of transport block set size is [0, 20000].  
Transport block set size must be an integer multiple of transport block size.

### 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>DataI</td>
<td>input data</td>
<td>real</td>
</tr>
<tr>
<td>2</td>
<td>TFI_I</td>
<td>transport format indicator</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>DataO</td>
<td>output data</td>
<td>real</td>
</tr>
<tr>
<td>4</td>
<td>TFI_O</td>
<td>transport format indicator</td>
<td>int</td>
</tr>
</tbody>
</table>

**Notes/Equations**

1. This model implements the reverse process of radio frame segmentation.

2. Each firing, 1 TFI_I token and N DataO tokens are produced when 1 TFI_I and N DataI tokens are consumed, while N is calculated using the maximum value of transport block set size, that is, the maximum valid data in one TTI.

3. The TFI value is an index used to select the transport block size and transport block set size from the transport format set, as specified by DynTF_Set. The minimum TFI is 0, the step is 1. The number of input and output tokens in each firing is calculated using the maximum value of transport block set size, while the valid tokens are calculated using the transport block set indexed by TFI.

4. When the transmission time interval is longer than 10 msec, the input bit sequence on consecutive F_I radio frames is combined and mapped onto one TTI.
References

**TDSCDMA_DeRateMatch**

Description: Derate match
Library: TDSCDMA, Multiplexing & Coding
Class: SDFTDSCDMA_DeRateMatch
Derived From: TDSCDMA_CCTrCH_MuxBase
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>PhyChNum_SA</td>
<td>physical channel allocation configuration</td>
<td>0 0 1 0 0 0 0</td>
<td>int array</td>
<td>[0, 2] for Uplink, [0, 16] for Downlink</td>
</tr>
<tr>
<td>TrChNum</td>
<td>number of Transport Channels</td>
<td>2</td>
<td>int</td>
<td>[1, 32]</td>
</tr>
<tr>
<td>RM_TA</td>
<td>rate matching attributes of all Transport Channels</td>
<td>1 1</td>
<td>int array</td>
<td>[1, 256]</td>
</tr>
<tr>
<td>DynTF_Set_TA</td>
<td>dynamic part of TF set of all Transport Channels</td>
<td>100 100 244 244</td>
<td>int array</td>
<td>[0, 5000] for transport block size, [0, 20000] for transport block set size</td>
</tr>
<tr>
<td>TF_SetSize_TA</td>
<td>transport format set size of all Transport Channels</td>
<td>1 1</td>
<td>int array</td>
<td>[1, 64] for each element</td>
</tr>
<tr>
<td>TTI_TA</td>
<td>transmission time interval of all Transport Channels</td>
<td>2 1</td>
<td>int array</td>
<td>[0, 3] for each element</td>
</tr>
<tr>
<td>CRC_TA</td>
<td>number of CRC bits of all Transport Channels</td>
<td>2 3</td>
<td>int array</td>
<td>[0, 4] for each element</td>
</tr>
<tr>
<td>ChCodingType_TA</td>
<td>channel coding type of all Transport Channels</td>
<td>2 2</td>
<td>int array</td>
<td>[0, 3] for each element</td>
</tr>
<tr>
<td>PuncLimit</td>
<td>puncturing limit</td>
<td>2/3</td>
<td>real</td>
<td>[0, 1]</td>
</tr>
<tr>
<td>Link</td>
<td>link selection: Downlink, Uplink</td>
<td></td>
<td>enum</td>
<td></td>
</tr>
<tr>
<td>ModType_SA</td>
<td>type of modulation of all slots, 0 for QPSK, 1 for 8PSK</td>
<td>0 0 0 0 0 0 0</td>
<td>int array</td>
<td>[0, 1] for each element</td>
</tr>
</tbody>
</table>
5-42 TDSCDMA_DeRateMatch

Multiplexing and Coding Components

<table>
<thead>
<tr>
<th>Pin Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>DataIn</td>
<td>input data</td>
<td>real</td>
</tr>
<tr>
<td>TFCI</td>
<td>transport channel combination indicator</td>
<td>int</td>
</tr>
</tbody>
</table>

Notes/Equations

1. This model is used to perform derate matching, the inverse operation of rate matching. Each firing, 1 token consumed at TFCI indicates the value of transport format combination indicator. The maximum number of rate-matched tokens in one frame for all transport channels involved in rate matching are consumed at DataIn and the possible maximum number of tokens in one frame.
for all transport formats of the transport channel specified by TrChIndex exported at DataOut each firing.

2. Derate matching means removing the repeated tokens which are added in the rate matching or insert zeros where the tokens are punctured in the rate matching.

3. Higher layers assign a rate-matching attribute for each transport channel. This attribute is semi-static and can only be changed through higher layer signaling. The rate-matching attribute is used when the number of bits to be repeated or punctured is calculated. RM_TA is provided so that users can set the semi-static attributes for each transport channel.

   The number of bits on a transport channel can vary between different transmission time intervals. When the number of bits between different transmission time intervals changes, bits are repeated or punctured to ensure that the total bit rate after TrCh multiplexing is the same as the total channel bit rate of the allocated physical channels.

   For rate matching algorithm details, refer to [2].

4. All transport channel information must be provided in the form of arrays.

   For DynTF_Set_TA the correct form is transport block size 1, transport block set size 1, transport block size 2, transport block set size 2, etc. The size of this array must be a multiple of 2, and the transport block set size must be a multiple of the relative transport block size.

   When setting TTI_TA, CRC_TA and ChCodingType_TA, refer to Table 5-4.

5. PuncLimit denotes the variable PL defined in [2]. Refer to [2] for details regarding use of this variable in rate matching algorithm.
Multiplexing and Coding Components

6. PhyChNum_SA indicates the number of allocated physical channels in each slot. The sum of PhyChNum_SA elements is the number of allocated physical channels.

7. TFCI_SA indicates in which slots TFCI bits will be transmitted. 0 denotes no TFCI bits will be transmitted in the slot. 1 denotes TFCI bits can be transmitted in the slot. The setting must be consistent with PhyChNum_SA setting, which means TFCI bits can only be transmitted in those slots in which the elements of PhyChNum_SA are not zero. Only the first allocated physical channel in each slot is used to transmit TFCI bits.

8. TFCI_Length_SA indicates the number of TFCI bits transmitted in each slot. If 0 is selected, the number of TFCI bits transmitted in the slot is dependent on the value of TFCI imported at TFCI pin. If a non-zero is selected, the number of TFCI bits transmitted in the slot is this non-zero value.

9. MinSF_PA indicates the minimum spread factor that can be used for corresponding physical channel. The size of MinSF_PA must be equal to the sum of PhyChNum_SA elements.

10. NdataOption specifies how the target number of rate-matched data is calculated with MinSF_PA. For Downlink, only Minimum can be selected. For Uplink, both can be selected. Refer to [2] for details.

11. SS_TPC_SA indicates the number of SS and TPC symbols transmitted in each slot. 1 denotes one SS and one TPC symbols are transmitted in the slot; 2 denotes no SS and no TPC symbols are transmitted in the slot; 3 denotes 16/SF SS and 16/SF TPC symbols are transmitted in the slot, where SF is the spreading factor of the physical channel used to transmitted SS and TPC symbols. Only the first allocated physical channel in each slot is used to transmit SS and TPC symbols.

References


TDSCDMA_DeSubFrameSeg

Description  Subframe desegmentation
Library  TDSCDMA, Multiplexing & Coding
Class  SDFTDSCDMA_DeSubFrameSeg
Derived From  TDSCDMA_CCTrCH_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>PhyChNum_SA</td>
<td>physical channel allocation configuration</td>
<td>0 0 0 0 0 0</td>
<td>int array</td>
<td>[0, 2] for Uplink, [0, 16] for Downlink</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>DataI</td>
<td>input data</td>
<td>multiple real</td>
</tr>
<tr>
<td>2</td>
<td>SizeInM</td>
<td>input data length</td>
<td>multiple 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>DataO</td>
<td>output data</td>
<td>multiple real</td>
</tr>
<tr>
<td>4</td>
<td>SizeOutM</td>
<td>output data length</td>
<td>multiple int</td>
</tr>
</tbody>
</table>

Notes/Equations

1. This model performs subframe desegmentation on subframes for each physical channel. Each firing, this model consumes $704 \times 3 \times 2$ tokens for each physical channel on multiple pin DataI, 2 tokens for each physical channel on multiple pin SizeInM, and exports $704 \times 3 \times 2$ tokens for each physical channel on...
Multiplexing and Coding Components

multiple pin DataO, 1 token for each physical channel on multiple pin SizeOutM.

2. PhyChNum_SA indicates the number of physical channels allocated in each time slot.

3. Each firing, two subframes constructing one frame are consumed on DataI for each physical channel, which contains 704 × 3 × 2 tokens, the maximum number of tokens one frame can contain for one physical channel. Each subframe consists of valid tokens and zero padding tokens. The number of valid tokens is specified by the token consumed on SizeInM. If the value of this token is L, the model combines the first L tokens of the first subframe with the first L tokens of the second subframe and exports them with 704 × 3 × 2-2 × L padding tokens followed on multiple pin DataO for each physical channel. And the number of valid tokens in one frame 2 × L is exported on multiple pin SizeOutM for each physical channel.

References

TDSCDMA_PhyChMap

Description  Physical channel mapping
Library    TDSCDMA, Multiplexing & Coding
Class     SDFTDSCDMA_PhyChMap
Derived From  TDSCDMA_CCTrCH_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>PhyChNum_SA</td>
<td>physical channel allocation</td>
<td>0 0 1 0 0 0 0</td>
<td>int array</td>
<td>[0, 2] for Uplink, [0, 16] for Downlink</td>
</tr>
<tr>
<td>Link</td>
<td>link selection: Downlink, Uplink</td>
<td>Downlink</td>
<td>enum</td>
<td></td>
</tr>
<tr>
<td>ModType_SA</td>
<td>type of modulation of all slots, 0 for QPSK, 1 for 8PSK</td>
<td>0 0 0 0 0 0 0</td>
<td>int array</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>DataI</td>
<td>input data</td>
<td>multiple int</td>
</tr>
<tr>
<td>2</td>
<td>SltFmtIn</td>
<td>input data slot format</td>
<td>multiple 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>DataO</td>
<td>output data</td>
<td>multiple int</td>
</tr>
<tr>
<td>4</td>
<td>SltFmtOut</td>
<td>output data slot format</td>
<td>multiple int</td>
</tr>
</tbody>
</table>

Notes/Equations
1. This model performs physical channel mapping. Each firing, this model consumes $704 \times 3$ tokens for each physical channel on multiple pin DataI, which is the maximum number of tokens one physical channel can contain in one subframe. 1 token for each physical channel on multiple pin SltFmtIn is consumed to indicate the slot format index for the physical channel. $704 \times 3$ mapped tokens are exported for each physical channel in one subframe on multiple pin DataO and 1 token is exported on multiple pin SltFmtOut to indicate the slot format index for the physical channel.

2. PhyChNum_SA indicates the number of physical channels allocated in each time slot.

3. For details regarding physical channel mapping algorithm, refer to [1].

References

TDSCDMA_PhyChSeg

Description  Physical channel segmentation
Library  TDSCDMA, Multiplexing & Coding
Class  SDFTDSCDMA_PhyChSeg
Derived From  TDSCDMA_CCTrCH_MuxBase
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>PhyChNum_SA</td>
<td>physical channel allocation configuration</td>
<td>0 0 1 0 0 0 0</td>
<td>int array</td>
<td>[0, 2] for Uplink, [0, 16] for Downlink</td>
</tr>
<tr>
<td>TrChNum</td>
<td>number of Transport Channels</td>
<td>2</td>
<td>int</td>
<td>[1, 32]</td>
</tr>
<tr>
<td>RM_TA</td>
<td>rate matching attributes of all Transport Channels</td>
<td>1 1</td>
<td>int array</td>
<td>[1, 256]</td>
</tr>
<tr>
<td>DynTF_Set_TA</td>
<td>dynamic part of TF set of all Transport Channels</td>
<td>100 100 244 244</td>
<td>int array</td>
<td>[0, 5000] for transport block size, [0, 20000] for transport block set size</td>
</tr>
<tr>
<td>TF_SetSize_TA</td>
<td>transport format set size of all Transport Channels</td>
<td>1 1</td>
<td>int array</td>
<td>[1, 64] for each element</td>
</tr>
<tr>
<td>TTI_TA</td>
<td>transmission time interval of all Transport Channels</td>
<td>2 1</td>
<td>int array</td>
<td>[0, 3] for each element</td>
</tr>
<tr>
<td>CRC_TA</td>
<td>number of CRC bits of all Transport Channels</td>
<td>2 3</td>
<td>int array</td>
<td>[0, 4] for each element</td>
</tr>
<tr>
<td>ChCodingType_TA</td>
<td>channel coding type of all Transport Channels</td>
<td>2 2</td>
<td>int array</td>
<td>[0, 3] for each element</td>
</tr>
<tr>
<td>PuncLimit</td>
<td>puncturing limit</td>
<td>2/3</td>
<td>real</td>
<td>[0, 1]</td>
</tr>
<tr>
<td>Link</td>
<td>link selection: Downlink, Uplink</td>
<td>Uplink</td>
<td>enum</td>
<td></td>
</tr>
<tr>
<td>ModType_SA</td>
<td>type of modulation of all slots, 0 for QPSK, 1 for 8PSK</td>
<td>0 0 0 0 0 0 0</td>
<td>int array</td>
<td>[0, 1] for each element</td>
</tr>
</tbody>
</table>
# Multiplexing and Coding Components

1. This model is used to perform physical channel segmentation.

Each firing, \( bm_{\text{PhyChNumAll}} \times \text{MAX_BIT_SLOT} \) tokens are consumed at DataIn, in which \( bm_{\text{PhyChNumAll}} \) is the number of allocated physical channels and MAX_BIT_SLOT is the possible maximum number of bits in one physical channel, that is \( 704 \times 3 \times 2 \); 1 token consumed at TFCI indicates the value of transport format combination indicator; MAX_BIT_SLOT tokens for each physical channel are exported at multiple pin DataOut which include valid tokens and padding tokens; 1 token is exported for each physical channel at multiple pin SizeOutM which indicates the number of valid tokens; 2 tokens are

---

### 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>DataIn</td>
<td>Input data</td>
<td>int</td>
</tr>
<tr>
<td>2</td>
<td>TFCI</td>
<td>transport format combination indicator</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>DataOut</td>
<td>output data</td>
<td>multiple int</td>
</tr>
<tr>
<td>4</td>
<td>SizeOutM</td>
<td>output data length</td>
<td>multiple int</td>
</tr>
<tr>
<td>5</td>
<td>SltFmtOut</td>
<td>slot format of each physical channel</td>
<td>multiple int</td>
</tr>
</tbody>
</table>

### Notes/Equations

1. This model is used to perform physical channel segmentation.
exported for each physical channel at multiple pin SltFmtOut which indicate the slot format of each physical channel.

2. Each firing, this model consumes one CCTrCH data block from the DataIn pin. When more than one physical channel is used, physical channel segmentation divides the CCTrCH data block among different physical channels.

The bits input to the physical channel segmentation are denoted by \( x_1, x_2, x_3, ..., x_Y \), where \( Y \) is the number of bits input to the physical channel segmentation block. The number of physical channels is denoted by \( P \). Bits after physical channel segmentation are denoted \( u_{p1}, u_{p2}, u_{p3}, ..., u_{p, U_p} \) where \( p \) is physical channel number and \( U_p \) is the number of bits in physical channel \( p \).

The relation between \( x_k \) and \( u_{pk} \) is given below:

- Bits on first physical channel after physical channel segmentation:
  \[ u_{1,k} = x_{i,k} = 1, 2, ..., U_1 \]

- Bits on second physical channel after physical channel segmentation:
  \[ u_{2,k} = x_{i,k+U} = 1, 2, ..., U_2 \]

- Bits on the \( P \)th physical channel after physical channel segmentation:
  \[ u_{P,k} = x_{i,k+(P-1)U} = 1, 2, ..., U_p \]

The resulting physical channels are exported at DataOut, if \( U_p(p=1,2,...,P) \) is less than MAX_BIT_SLOT, padding bits(0) are added.

3. All transport channel information must be provided in the form of arrays.

For DynTF_Set_TA the correct form is transport block size 1, transport block set size 1, transport block size 2, transport block set size 2, etc. The size of this array must be a multiple of 2, and the transport block set size must be a multiple of the relative transport block size.

When setting TTI_TA, CRC_TA and ChCodingType_TA, refer to Table 5-5.

<table>
<thead>
<tr>
<th>Table 5-5. Array Values</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>TTI_TA</strong></td>
</tr>
<tr>
<td>TTI</td>
</tr>
<tr>
<td>Time</td>
</tr>
<tr>
<td>10ms</td>
</tr>
<tr>
<td>20ms</td>
</tr>
</tbody>
</table>

TDSCDMA_PhysicsSeg 5-51
4. PuncLimit denotes the variable PL defined in [2]. Refer to [2] for details regarding use of this variable in rate matching algorithm.

5. PhyChNum_SA indicates the number of allocated physical channels in each slot. The sum of PhyChNum_SA elements is the number of allocated physical channels.

6. TFCI_SA indicates in which slots TFCI bits will be transmitted. 0 denotes no TFCI bits will be transmitted in the slot. 1 denotes TFCI bits can be transmitted in the slot. The setting must be consistent with PhyChNum_SA setting, which means TFCI bits can only be transmitted in those slots in which the elements of PhyChNum_SA are not zero. Only the first allocated physical channel in each slot is used to transmit TFCI bits.

7. TFCI_Length_SA indicates the number of TFCI bits transmitted in each slot. If 0 is selected, the number of TFCI bits transmitted in the slot is dependent on the value of TFCI imported at TFCI pin. If a non-zero is selected, the number of TFCI bits transmitted in the slot is this non-zero value.

8. MinSF_PA indicates the minimum spread factor that can be used for corresponding physical channel. The size of MinSF_PA must be equal to the sum of PhyChNum_SA elements.

9. NdataOption specifies how the target number of rate-matched data is calculated with MinSF_PA. For Downlink, only Minimum can be selected. For Uplink, both can be selected. Refer to [2] for details.

10. SS_TPC_SA indicates the number of SS and TPC symbols transmitted in each slot. 1 denotes one SS and one TPC symbols are transmitted in the slot; 2 denotes no SS and no TPC symbols are transmitted in the slot; 3 denotes 16/SF SS and 16/SF TPC symbols are transmitted in the slot, where SF is the spreading factor of the physical channel used to transmitted SS and TPC.

---

Table 5-5. Array Values

<table>
<thead>
<tr>
<th>TTI_TA</th>
<th>CRC_TA</th>
<th>ChCodingType_TA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time</td>
<td>Value</td>
<td>Coding</td>
</tr>
<tr>
<td>40ms</td>
<td>2</td>
<td>12 bits</td>
</tr>
<tr>
<td>80ms</td>
<td>3</td>
<td>16 bits</td>
</tr>
<tr>
<td></td>
<td>24 bits</td>
<td>4</td>
</tr>
</tbody>
</table>

CC = convolution coding; TC = turbo coding.

---
symbols. Only the first allocated physical channel in each slot is used to transmit SS and TPC symbols.

References


Multiplexing and Coding Components

TDSCDMA_RadioEqual

Description  Radio frame size equalization
Library   TDSCDMA, Multiplexing & Coding
Class   SDFTDSCDMA_RadioEqual
Derived From  TDSCDMA_ChEncodingBase
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>DynTF_Set</td>
<td>dynamic part of Transport Format Set</td>
<td>244 488</td>
<td>int array</td>
<td>†</td>
</tr>
<tr>
<td>TTI</td>
<td>transmission time interval: TTI_10ms, TTI_20ms, TTI_40ms, TTI_80ms</td>
<td>TTI_10ms</td>
<td>enum</td>
<td></td>
</tr>
<tr>
<td>CRC</td>
<td>length of CRC bits: No_CRC, CRC_8_bits, CRC_12_bits, CRC_16_bits, CRC_24_bits</td>
<td>CRC_16_bits</td>
<td>enum</td>
<td></td>
</tr>
<tr>
<td>ChCodingType</td>
<td>channel coding type: No_Coding, CC_HalfRate, CC_OneThirdRate, TurboCoding</td>
<td>CC_HalfRate</td>
<td>enum</td>
<td></td>
</tr>
</tbody>
</table>

† The array structure of DynTF_Set is [transport block size 1, transport block set size 1, transport block size 2, transport block set size 2, ...].

The value range of transport block size is [0, 5000].
The value range of transport block set size is [0, 20000].
Transport block set size must be an integer multiple of transport block size.

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>DataI</td>
<td>transport block set</td>
<td>int</td>
</tr>
<tr>
<td>2</td>
<td>TFI_I</td>
<td>transport format indicator</td>
<td>int</td>
</tr>
</tbody>
</table>
Notes/Equations

1. This model implements radio frame size equalization.

   Each firing, 1 TFI_I token and N DataO tokens are produced when 1 TFI_I and M DataI tokens are consumed, while N and M are calculated using the maximum value of transport block set size; that is, the maximum valid data in one TTI before and after radio frame size equalization.

   The value of TFI is an index used to select the transport block size and transport block set size from the transport format set, as specified by DynTF_Set. The minimum TFI is 0, the step is 1. The number of input and output tokens in each firing is calculated using the maximum value of transport block set size, while the valid tokens are calculated using the transport block set indexed by TFI.

2. Radio frame size equalization is padding the input bit sequence in order to ensure that the output can be averaged into radio frames if the number of radio frames in one TTI is larger than 1.

References

TDSCDMA_RadioSeg

Description  Radio frame segmentation
Library  TDSCDMA, Multiplexing & Coding
Class  SDFTDSCDMA_RadioSeg
Derived From  TDSCDMA_ChEncodingBase
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>DynTF_Set</td>
<td>dynamic part of Transport Format Set</td>
<td>244 488</td>
<td>int array</td>
<td></td>
</tr>
<tr>
<td>TTI</td>
<td>transmission time interval: TTI_10ms, TTI_20ms, TTI_40ms, TTI_80ms</td>
<td>TTI_10ms</td>
<td>enum</td>
<td></td>
</tr>
<tr>
<td>CRC</td>
<td>length of CRC bits: No_CRC, CRC_8_bits, CRC_12_bits, CRC_16_bits, CRC_24_bits</td>
<td>CRC_16_bits</td>
<td>enum</td>
<td></td>
</tr>
<tr>
<td>ChCodingType</td>
<td>channel coding type: No_Coding, CC_HalfRate, CC_OneThirdRate, TurboCoding</td>
<td>CC_HalfRate</td>
<td>enum</td>
<td></td>
</tr>
</tbody>
</table>

† The array structure of DynTF_Set is [transport block size 1, transport block set size 1, transport block size 2, transport block set size 2, ...].
The value range of transport block size is [0, 5000].
The value range of transport block set size is [0, 20000].
Transport block set size must be an integer multiple of transport block size.

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>DataI</td>
<td>transport block set</td>
<td>int</td>
</tr>
<tr>
<td>2</td>
<td>TFI_I</td>
<td>transport format indicator</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>DataO</td>
<td>transport block set with CRC attached</td>
<td>int</td>
</tr>
<tr>
<td>4</td>
<td>TFI_O</td>
<td>transport format indicator</td>
<td>int</td>
</tr>
</tbody>
</table>

Notes/Equations

1. This model implements radio frame segmentation.

   Each firing, 1 TFI_I token and N DataO tokens are produced when 1 TFI_I and N DataI tokens consumed, while N is calculated using the maximum value of transport block set size, that is, the maximum valid data in one TTI.

   TFI value is an index used to select the transport block size and transport block set size from the transport format set, as specified by DynTF_Set. The minimum TFI is 0, the step is 1. The number of input and output tokens in each firing is calculated using the maximum value of transport block set size, while the valid tokens are calculated using the transport block set indexed by TFI.

2. When the transmission time interval is longer than 10 ms, the input bit sequence is segmented and mapped onto consecutive $F_t$ radio frames. Following radio frame size equalization the input bit sequence length is guaranteed to be an integer multiple of $F_t$.

References

TDSCDMA_RateMatch

Description  Rate match
Library  TDSCDMA, Multiplexing & Coding
Class  SDFTDSCDMA_RateMatch
Derived From  TDSCDMA_CCTrCH_MuxBase
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>PhyChNum_SA</td>
<td>physical channel allocation configuration</td>
<td>0 0 1 0 0 0 0</td>
<td>int array</td>
<td>[0, 2] for Uplink, [0, 16] for Downlink</td>
</tr>
<tr>
<td>TrChNum</td>
<td>number of Transport Channels</td>
<td>2</td>
<td>int</td>
<td>[1, 32]</td>
</tr>
<tr>
<td>RM_T A</td>
<td>rate matching attributes of all Transport Channels</td>
<td>1 1</td>
<td>int array</td>
<td>[1, 256]</td>
</tr>
<tr>
<td>DynTF_Set_T A</td>
<td>dynamic part of TF set of all Transport Channels</td>
<td>100 100 244 244</td>
<td>int array</td>
<td>[0, 5000] for transport block size, [0, 20000] for transport block set size</td>
</tr>
<tr>
<td>TF_SetSize_T A</td>
<td>transport format set size of all Transport Channels</td>
<td>1 1</td>
<td>int array</td>
<td>[1, 64] for each element</td>
</tr>
<tr>
<td>TTI_T A</td>
<td>transmission time interval of all Transport Channels</td>
<td>2 1</td>
<td>int array</td>
<td>[0, 3] for each element</td>
</tr>
<tr>
<td>CRC_T A</td>
<td>number of CRC bits of all Transport Channels</td>
<td>2 3</td>
<td>int array</td>
<td>[0, 4] for each element</td>
</tr>
<tr>
<td>ChCodingType_T A</td>
<td>channel coding type of all Transport Channels</td>
<td>2 2</td>
<td>int array</td>
<td>[0, 3] for each element</td>
</tr>
<tr>
<td>PuncLimit</td>
<td>puncturing limit</td>
<td>2/3</td>
<td>real</td>
<td>[0, 1]</td>
</tr>
<tr>
<td>Link</td>
<td>link selection: Downlink, Uplink</td>
<td>Uplink</td>
<td>enum</td>
<td></td>
</tr>
<tr>
<td>ModType_SA</td>
<td>type of modulation of all slots, 0 for QPSK, 1 for 8PSK</td>
<td>0 0 0 0 0 0 0</td>
<td>int array</td>
<td>[0, 1] for each element</td>
</tr>
</tbody>
</table>
1. This model is used to perform rate matching.

   Each firing, 1 token consumed at TFCI indicates the value of transport format combination indicator. The maximum number of tokens possible in one frame for all transport formats of the channel specified by TrChIndex are consumed at
DataIn; the maximum number of rate-matched tokens in one frame for all transport channels involved in rate matching are exported at DataOut.

2. Rate matching means that bits on a transport channel are repeated or punctured. Higher layers assign a rate-matching attribute for each transport channel. This attribute is semi-static and can only be changed through higher layer signaling. The rate-matching attribute is used when the number of bits to be repeated or punctured is calculated. RM_TA is provided so that users can set the semi-static attributes for each transport channel.

The number of bits on a transport channel can vary between different transmission time intervals. When the number of bits between different transmission time intervals changes, bits are repeated or punctured to ensure that the total bit rate after TrCh multiplexing is the same as the total channel bit rate of the allocated physical channels.

For rate matching algorithm details, refer to [2].

3. All transport channel information must be provided in the form of arrays.

For DynTF_Set_TA the correct form is transport block size 1, transport block set size 1, transport block size 2, transport block set size 2, etc. The size of this array must be a multiple of 2, and the transport block set size must be a multiple of the relative transport block size.

When setting TTI_TA, CRC_TA and ChCodingType_TA, refer to Table 5-6.

<table>
<thead>
<tr>
<th>TTI_TA</th>
<th>CRC_TA</th>
<th>ChCodingType_TA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time</td>
<td>Value</td>
<td>Coding</td>
</tr>
<tr>
<td>10ms</td>
<td>0</td>
<td>No CRC</td>
</tr>
<tr>
<td>20ms</td>
<td>1</td>
<td>8 bits</td>
</tr>
<tr>
<td>40ms</td>
<td>2</td>
<td>12 bits</td>
</tr>
<tr>
<td>80ms</td>
<td>3</td>
<td>16 bits</td>
</tr>
<tr>
<td></td>
<td></td>
<td>24 bits</td>
</tr>
</tbody>
</table>

CC = convolutional coding; TC = turbo coding

4. PuncLimit denotes the variable PL defined in [2]. Refer to [2] for details regarding use of this variable in rate matching algorithm.
5. PhyChNum_SA indicates the number of allocated physical channels in each slot. The sum of PhyChNum_SA elements is the number of allocated physical channels.

6. TFCI_SA indicates in which slots TFCI bits will be transmitted. 0 denotes no TFCI bits will be transmitted in the slot. 1 denotes TFCI bits can be transmitted in the slot. The setting must be consistent with PhyChNum_SA setting, which means TFCI bits can only be transmitted in those slots in which the elements of PhyChNum_SA are not zero. Only the first allocated physical channel in each slot is used to transmit TFCI bits.

7. TFCI_Length_SA indicates the number of TFCI bits transmitted in each slot. If 0 is selected, the number of TFCI bits transmitted in the slot is dependent on the value of TFCI imported at TFCI pin. If a non-zero is selected, the number of TFCI bits transmitted in the slot is this non-zero value.

8. MinSF_PA indicates the minimum spread factor that can be used for corresponding physical channel. The size of MinSF_PA must be equal to the sum of PhyChNum_SA elements.

9. NdataOption specifies how the target number of rate-matched data is calculated with MinSF_PA. For Downlink, only Minimum can be selected. For Uplink, both can be selected. Refer to [2] for details.

10. SS_TPC_SA indicates the number of SS and TPC symbols transmitted in each slot. 1 denotes one SS and one TPC symbols are transmitted in the slot; 2 denotes no SS and no TPC symbols are transmitted in the slot; 3 denotes 16/SF SS and 16/SF TPC symbols are transmitted in the slot, where SF is the spreading factor of the physical channel used to transmitted SS and TPC symbols. Only the first allocated physical channel in each slot is used to transmit SS and TPC symbols.

References


### TDSCDMA_RefChDecoder

**Description**  
TDSCDMA reference measurement channel decoder

**Library**  
TDSCDMA, Multiplexing & 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>Link</td>
<td>link selection: Downlink, Uplink</td>
<td>Downlink</td>
<td>enum</td>
<td></td>
</tr>
<tr>
<td>RefCh</td>
<td>reference channel selection indicator: CH_12.2k_MultiCode, CH_12.2k_SingleCode, CH_64k, CH_144k, CH_384k</td>
<td>CH_12.2k_MultiCode</td>
<td>enum</td>
<td></td>
</tr>
<tr>
<td>PhyChNum_SA</td>
<td>physical channel allocation configuration</td>
<td>0 0 2 0 0 0 0</td>
<td>int array</td>
<td>[0, 16] for Downlink, [0,2] for Uplink</td>
</tr>
<tr>
<td>MaxPhyChNum</td>
<td>sum of allocated physical channel in all slots</td>
<td>2</td>
<td>int</td>
<td>[1, 112]</td>
</tr>
<tr>
<td>ModType_SA</td>
<td>type of modulation of all slots, 0 for QPSK, 1 for 8PSK</td>
<td>0 0 0 0 0 0</td>
<td>int array</td>
<td>[0, 1]</td>
</tr>
<tr>
<td>TFCI_SA</td>
<td>allocated TFCI transmitted active slots configuration</td>
<td>0 0 1 0 0 0 0</td>
<td>int array</td>
<td>[0, 1]</td>
</tr>
<tr>
<td>TFCI_Length_SA</td>
<td>length of TFCI of all slots</td>
<td>0 0 16 0 0 0 0</td>
<td>int array</td>
<td>[0, 4,8,16,32] for QPSK, [0,6,12,24,48] for 8PSK</td>
</tr>
<tr>
<td>SS_TPC_SA</td>
<td>type of SS and TPC of all slots</td>
<td>2 2 1 2 2 2 2</td>
<td>int array</td>
<td>[1, 2,3]</td>
</tr>
<tr>
<td>MinSF_PA</td>
<td>minimum spreading factor array corresponding to allocated physical channels</td>
<td>16 16</td>
<td>int array</td>
<td>[1, 16] for Downlink, [1,2,4,8,16] for Uplink</td>
</tr>
</tbody>
</table>

5-62  
TDSCDMA_RefChDecoder
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>DataI</td>
<td>output data</td>
<td>multiple real</td>
</tr>
<tr>
<td>2</td>
<td>SizeInM</td>
<td>output data length</td>
<td>multiple 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>DCH</td>
<td>DCH data out</td>
<td>int</td>
</tr>
</tbody>
</table>

Notes/Equations

1. This subnetwork implements reference measurement channel. The schematic for this subnetwork is shown in Figure 5-1.

2. MinSF_PA indicates the minimum spread factor that can be used for corresponding physical channel. The size of MinSF_PA must be equal to the sum of PhyChNum_SA elements.

3. SS_TPC_SA indicates the number of SS and TPC symbols transmitted in each slot. 1 denotes one SS and one TPC symbols are transmitted in the slot; 2 denotes no SS and no TPC symbols are transmitted in the slot; 3 denotes 16/SF
SS and 16/SF TPC symbols are transmitted in the slot, where SF is the spreading factor of the physical channel used to transmitted SS and TPC symbols. Only the first allocated physical channel in each slot is used to transmit SS and TPC symbols.

4. The structure and settings for different data rates are given in Table 5-7 through Table 5-14.

### Table 5-7. 12.2 kbps UL Reference Measurement Channel

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Information data rate</td>
<td>12.2 kbps</td>
</tr>
<tr>
<td>RU’s allocated</td>
<td>1TS (1 × SF8) = 2RU/5ms</td>
</tr>
<tr>
<td>Midamble</td>
<td>144</td>
</tr>
<tr>
<td>Interleaving</td>
<td>20 ms</td>
</tr>
<tr>
<td>Power control</td>
<td>4 Bit/user/10ms</td>
</tr>
<tr>
<td>TFCI</td>
<td>16 Bit/user/10ms</td>
</tr>
<tr>
<td>4 Bit reserved for future use (place of SS)</td>
<td>4 Bit/user/10ms</td>
</tr>
<tr>
<td>Inband signalling DCCH</td>
<td>2.4 kbps</td>
</tr>
<tr>
<td>Puncturing level at Code rate 1/3: DCH / DCCH</td>
<td>33% / 33%</td>
</tr>
</tbody>
</table>

### Table 5-8. 12.2 kbps UL and DL Multi-Code Reference Measurement Channel

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Information data rate</td>
<td>12.2 kbps</td>
</tr>
<tr>
<td>RU’s allocated</td>
<td>1TS (2 × SF16) = 2RU/5ms</td>
</tr>
<tr>
<td>Midamble</td>
<td>144</td>
</tr>
<tr>
<td>Interleaving</td>
<td>20 ms</td>
</tr>
<tr>
<td>Power control (TPC)</td>
<td>4 Bit/user/10ms</td>
</tr>
<tr>
<td>TFCI</td>
<td>16 Bit/user/10ms</td>
</tr>
<tr>
<td>Synchronization Shift (SS)</td>
<td>4 Bit/user/10ms</td>
</tr>
<tr>
<td>Inband signalling DCCH</td>
<td>2.4 kbps</td>
</tr>
<tr>
<td>Puncturing level at Code rate 1/3: DCH / DCCH</td>
<td>33% / 33%</td>
</tr>
</tbody>
</table>
### Table 5-9. 64 kbps UL Reference Measurement Channel

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Information data rate</td>
<td>64 kbps</td>
</tr>
<tr>
<td>RU's allocated</td>
<td>1TS (1 × SF2) = 8RU/5ms</td>
</tr>
<tr>
<td>Midamble</td>
<td>144</td>
</tr>
<tr>
<td>Interleaving</td>
<td>20 ms</td>
</tr>
<tr>
<td>Power control (TPC)</td>
<td>4 Bit/user/10ms</td>
</tr>
<tr>
<td>TFCI</td>
<td>16 Bit/user/10ms</td>
</tr>
<tr>
<td>Synchronization Shift (SS)</td>
<td>4 Bit/user/10ms</td>
</tr>
<tr>
<td>Inband signalling DCCH</td>
<td>2.4 kbps</td>
</tr>
<tr>
<td>Puncturing level at Code rate: 1/3 DCH / ∫ DCCH</td>
<td>32% / 0</td>
</tr>
</tbody>
</table>

### Table 5-10. 64 kbps DL Reference Measurement Channel

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Information data rate</td>
<td>64 kbps</td>
</tr>
<tr>
<td>RU's allocated</td>
<td>1TS (8 × SF16) = 8RU/5ms</td>
</tr>
<tr>
<td>Midamble</td>
<td>144</td>
</tr>
<tr>
<td>Interleaving</td>
<td>20 ms</td>
</tr>
<tr>
<td>Power control (TPC)</td>
<td>4 Bit/user/10ms</td>
</tr>
<tr>
<td>TFCI</td>
<td>16 Bit/user/10ms</td>
</tr>
<tr>
<td>Synchronization Shift (SS)</td>
<td>4 Bit/user/10ms</td>
</tr>
<tr>
<td>Inband signalling DCCH</td>
<td>2.4 kbps</td>
</tr>
<tr>
<td>Puncturing level at Code rate: 1/3 DCH / ∫ DCCH</td>
<td>32% / 0</td>
</tr>
</tbody>
</table>
### Table 5-11. 144 kbps UL Reference Measurement Channel

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Information data rate</td>
<td>144 kbps</td>
</tr>
<tr>
<td>RU's allocated</td>
<td>$2TS \times (1 \times SF2) = 16RU/5ms$</td>
</tr>
<tr>
<td>Midamble</td>
<td>144</td>
</tr>
<tr>
<td>Interleaving</td>
<td>20 ms</td>
</tr>
<tr>
<td>Power control (TPC)</td>
<td>8 Bit/user/10ms</td>
</tr>
<tr>
<td>TFCI</td>
<td>32 Bit/user/10ms</td>
</tr>
<tr>
<td>Synchronization Shift (SS)</td>
<td>8 Bit/user/10ms</td>
</tr>
<tr>
<td>Inband signalling DCCH</td>
<td>2.4 kbps</td>
</tr>
<tr>
<td>Puncturing level at Code rate: $1/3$ DCH / DCCH</td>
<td>38% / 7%</td>
</tr>
</tbody>
</table>
5. The configuration for transport channels is fixed when the parameter Link and RefCh are set. The configuration for physical channels can be set flexibly according to Table 5-7 through Table 5-14. However, the settings for this subnetwork must match the settings of TDSCDMA_RefCh if it is used to transmit. An example for each configuration is shown in Table 5-15 through Table 5-22.

### Table 5-13. 384 kbps UL Reference Measurement Channel

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Information data rate</td>
<td>384 kbps</td>
</tr>
<tr>
<td>RU's allocated</td>
<td>4TS (1 × SF2 + 1 × SF8) = 40RU/5ms</td>
</tr>
<tr>
<td>Midamble</td>
<td>144</td>
</tr>
<tr>
<td>Interleaving</td>
<td>20 ms</td>
</tr>
<tr>
<td>Power control (TPC)</td>
<td>16 Bit/user/10ms</td>
</tr>
<tr>
<td>TFCI</td>
<td>64 Bit/user/10ms</td>
</tr>
<tr>
<td>Synchronization Shift (SS)</td>
<td>16 Bit/user/10ms</td>
</tr>
<tr>
<td>Inband signalling DCCH</td>
<td>max 2.0 kbps</td>
</tr>
<tr>
<td>Puncturing level at Code rate: 1/3 DCH / DCCH</td>
<td>41% / 12%</td>
</tr>
</tbody>
</table>

### Table 5-14. 384 kbps DL Reference Measurement Channel

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Information data rate</td>
<td>384 kbps</td>
</tr>
<tr>
<td>RU's allocated</td>
<td>4TS (10 × SF16) = 40RU/5ms</td>
</tr>
<tr>
<td>Midamble</td>
<td>144</td>
</tr>
<tr>
<td>Interleaving</td>
<td>20 ms</td>
</tr>
<tr>
<td>Power control (TPC)</td>
<td>16 Bit/user/10ms</td>
</tr>
<tr>
<td>TFCI</td>
<td>64 Bit/user/10ms</td>
</tr>
<tr>
<td>Synchronization Shift (SS)</td>
<td>16 Bit/user/10ms</td>
</tr>
<tr>
<td>Inband signalling DCCH</td>
<td>max 2 kbps</td>
</tr>
<tr>
<td>Puncturing level at Code rate: 1/3 DCH / DCCH</td>
<td>41% / 12%</td>
</tr>
</tbody>
</table>
### Table 5-15. Physical Channel Setting of 12.2 kbps UL Reference Measurement Channel

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Link</td>
<td>Uplink</td>
</tr>
<tr>
<td>RefCh</td>
<td>12.2K_SingleCode</td>
</tr>
<tr>
<td>PhyChNum_SA</td>
<td>0 0 1 0 0 0 0</td>
</tr>
<tr>
<td>MaxPhyChNum</td>
<td>1</td>
</tr>
<tr>
<td>ModType_SA</td>
<td>0[7]</td>
</tr>
<tr>
<td>TFCI_SA</td>
<td>0 0 1 0 0 0 0</td>
</tr>
<tr>
<td>TFCI_Length_SA</td>
<td>0 0 16 0 0 0 0</td>
</tr>
<tr>
<td>SS_TPC_SA</td>
<td>2 2 1 2 2 2 2</td>
</tr>
<tr>
<td>MinSF_PA</td>
<td>8</td>
</tr>
</tbody>
</table>

### Table 5-16. Physical Channel Setting of 12.2 kbps UL and DL Multi-Code Reference Measurement Channel

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Link</td>
<td>Uplink for UL and Downlink for DL</td>
</tr>
<tr>
<td>RefCh</td>
<td>12.2K_MultiCode</td>
</tr>
<tr>
<td>PhyChNum_SA</td>
<td>0 0 2 0 0 0 0</td>
</tr>
<tr>
<td>MaxPhyChNum</td>
<td>2</td>
</tr>
<tr>
<td>ModType_SA</td>
<td>0[7]</td>
</tr>
<tr>
<td>TFCI_SA</td>
<td>0 0 1 0 0 0 0</td>
</tr>
<tr>
<td>TFCI_Length_SA</td>
<td>0 0 16 0 0 0 0</td>
</tr>
<tr>
<td>SS_TPC_SA</td>
<td>2 2 1 2 2 2 2</td>
</tr>
<tr>
<td>MinSF_PA</td>
<td>16[2]</td>
</tr>
</tbody>
</table>
### Table 5-17. Physical Channel Setting
64 kbps UL Reference Measurement Channel

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Link</td>
<td>Uplink</td>
</tr>
<tr>
<td>RefCh</td>
<td>64K</td>
</tr>
<tr>
<td>PhyChNum_SA</td>
<td>0 0 1 0 0 0 0</td>
</tr>
<tr>
<td>MaxPhyChNum</td>
<td>1</td>
</tr>
<tr>
<td>ModType_SA</td>
<td>0[7]</td>
</tr>
<tr>
<td>TFCI_SA</td>
<td>0 0 1 0 0 0 0</td>
</tr>
<tr>
<td>TFCI_Length_SA</td>
<td>0 0 1 6 0 0 0 0</td>
</tr>
<tr>
<td>SS_TPC_SA</td>
<td>2 2 1 2 2 2 2</td>
</tr>
<tr>
<td>MinSF_PA</td>
<td>2</td>
</tr>
</tbody>
</table>

### Table 5-18. Physical Channel Setting of
64 kbps DL Reference Measurement Channel

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Link</td>
<td>Downlink</td>
</tr>
<tr>
<td>RefCh</td>
<td>64K</td>
</tr>
<tr>
<td>PhyChNum_SA</td>
<td>0 0 8 0 0 0 0</td>
</tr>
<tr>
<td>MaxPhyChNum</td>
<td>8</td>
</tr>
<tr>
<td>ModType_SA</td>
<td>0[7]</td>
</tr>
<tr>
<td>TFCI_SA</td>
<td>0 0 1 0 0 0 0</td>
</tr>
<tr>
<td>TFCI_Length_SA</td>
<td>0 0 1 6 0 0 0 0</td>
</tr>
<tr>
<td>SS_TPC_SA</td>
<td>2 2 1 2 2 2 2</td>
</tr>
<tr>
<td>MinSF_PA</td>
<td>16[8]</td>
</tr>
</tbody>
</table>
### Table 5-19. Physical Channel Setting of 144 kbps UL Reference Measurement Channel

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Link</td>
<td>Uplink</td>
</tr>
<tr>
<td>RefCh</td>
<td>144K</td>
</tr>
<tr>
<td>PhyChNum_SA</td>
<td>0 1 1 0 0 0</td>
</tr>
<tr>
<td>MaxPhyChNum</td>
<td>2</td>
</tr>
<tr>
<td>ModType_SA</td>
<td>0[7]</td>
</tr>
<tr>
<td>TFCI_SA</td>
<td>0 1 1 0 0 0</td>
</tr>
<tr>
<td>TFCI_Length_SA</td>
<td>0 0 16 16 0 0</td>
</tr>
<tr>
<td>SS_TPC_SA</td>
<td>2 2 1 2 2 2</td>
</tr>
<tr>
<td>MinSF_PA</td>
<td>2[2]</td>
</tr>
</tbody>
</table>

### Table 5-20. Physical Channel Setting of 144 kbps DL Reference Measurement Channel

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Link</td>
<td>Downlink</td>
</tr>
<tr>
<td>RefCh</td>
<td>144K</td>
</tr>
<tr>
<td>PhyChNum_SA</td>
<td>0 0 8 8 0 0</td>
</tr>
<tr>
<td>MaxPhyChNum</td>
<td>16</td>
</tr>
<tr>
<td>ModType_SA</td>
<td>0[7]</td>
</tr>
<tr>
<td>TFCI_SA</td>
<td>0 1 1 0 0 0</td>
</tr>
<tr>
<td>TFCI_Length_SA</td>
<td>0 0 16 16 0 0</td>
</tr>
<tr>
<td>SS_TPC_SA</td>
<td>2 2 1 2 2 2</td>
</tr>
<tr>
<td>MinSF_PA</td>
<td>16[16]</td>
</tr>
</tbody>
</table>
Table 5-21. Physical Channel Setting of 384 kbps UL Reference Measurement Channel

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Link</td>
<td>Uplink</td>
</tr>
<tr>
<td>RefCh</td>
<td>384K</td>
</tr>
<tr>
<td>PhyChNum_SA</td>
<td>0 0 2 2 2 0</td>
</tr>
<tr>
<td>MaxPhyChNum</td>
<td>8</td>
</tr>
<tr>
<td>ModType_SA</td>
<td>0[7]</td>
</tr>
<tr>
<td>TFCI_SA</td>
<td>0 0 1 1 1 0</td>
</tr>
<tr>
<td>TFCI_Length_SA</td>
<td>0 0 16 16 16 0</td>
</tr>
<tr>
<td>SS_TPC_SA</td>
<td>2 2 1 1 1 2</td>
</tr>
<tr>
<td>MinSF_PA</td>
<td>8 2 8 2 8 2</td>
</tr>
</tbody>
</table>

Table 5-22. Physical Channel Setting of 384 kbps DL Reference Measurement Channel

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Link</td>
<td>Downlink</td>
</tr>
<tr>
<td>RefCh</td>
<td>384K</td>
</tr>
<tr>
<td>PhyChNum_SA</td>
<td>0 0 1 0 1 0 1 0 0</td>
</tr>
<tr>
<td>MaxPhyChNum</td>
<td>1</td>
</tr>
<tr>
<td>ModType_SA</td>
<td>0[7]</td>
</tr>
<tr>
<td>TFCI_SA</td>
<td>0 0 1 1 1 0</td>
</tr>
<tr>
<td>TFCI_Length_SA</td>
<td>0 0 16 16 16 0</td>
</tr>
<tr>
<td>SS_TPC_SA</td>
<td>2 2 1 1 1 2</td>
</tr>
<tr>
<td>MinSF_PA</td>
<td>1 6[40]</td>
</tr>
</tbody>
</table>

References


TDSCDMA_RM_Cal

Description  TDSCDMA RM calculator
Library   TDSCDMA, Multiplexing & Coding
Class   SDFTDSCDMA_RM_Cal
Required Licenses

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Default</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>TrChNum</td>
<td>number of Transport Channels</td>
<td>2</td>
<td>int</td>
</tr>
<tr>
<td>RadioFrameSize_TA</td>
<td>radio frame size array</td>
<td>402 90</td>
<td>int array</td>
</tr>
<tr>
<td>MatchedSize_TA</td>
<td>matched radio frame size array</td>
<td>268 60</td>
<td>int array</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>RM</td>
<td>rate match attributor</td>
<td>int</td>
</tr>
<tr>
<td>2</td>
<td>PL</td>
<td>puncture limit</td>
<td>real</td>
</tr>
</tbody>
</table>

Notes/Equations

1. This model is used to calculate the semi-static rate matching attribute for each transport channel and puncturing limit.

2. Each firing, 1 PL token and TrChNum RM tokens are produced, where TrChNum is the number of transport channels.

3. RadioFrameSize_TA specifies the frame size of each transport channel before rate match.

4. MatchedSize_TA specifies the frame size of each transport channel after rate match.
References

Multiplexing and Coding Components

**TDSCDMA_SubFrameSeg**

Description  Subframe segmentation  
Library  TDSCDMA, Multiplexing & Coding  
Class  SDFTDSCDMA_SubFrameSeg  
Derived From  TDSCDMA_CCTrCH_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>PhyChNum_SA</td>
<td>physical channel allocation configuration</td>
<td>0 0 1 0 0 0</td>
<td>int array</td>
<td>[0, 2] for Uplink, [0, 16] for Downlink</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>DataI</td>
<td>input data</td>
<td>multiple int</td>
</tr>
<tr>
<td>2</td>
<td>SizeInM</td>
<td>input data length</td>
<td>multiple 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>DataO</td>
<td>output data</td>
<td>multiple int</td>
</tr>
</tbody>
</table>

**Notes/Equations**

1. This model performs subframe segmentation on frames for each physical channel. Each firing, this model consumes $704 \times 3 \times 2$ tokens from each physical channel on multiple pin DataI, 1 token for each physical channel on multiple pin SizeInM, and exports $704 \times 3 \times 2$ tokens for each physical channel on multiple pin DataO.
2. PhyChNum_SA indicates the number of physical channels allocated in each time slot.

3. Each firing, one frame of data bits are consumed on Data1 for each physical channel, which contains $704 \times 3 \times 2$ bits, the maximum number of data bits one frame can contain for one physical channel. The first part of frame are valid bits and the second are zero padding bits. The number of valid bits is specified by the token consumed on SizeInM. If the value of this token is $2 \times L$, the model divides the valid bits into two parts equally which has a length L each. The first L valid bits are exported with $(704 \times 3-L)$ padding bits, which form the first subframe; the second L valid bits are then exported with $(704 \times 3-L)$ padding bits, which form the second subframe.

References

TDSCDMA_TFCI_Encoder

**Description**  
TFCI coding for 1.28Mcps TDD

**Library**  
TDSCDMA, Multiplexing & Coding

**Class**  
SDFTDSCDMA_TFCI_Encoder

**Parameters**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Default</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>TFCICoding</td>
<td>TFCI coding option: QPSK_RM_32_10, QPSK_RM_16_5, QPSK_RP_4, QPSK_RP_8, 8PSK_RM_48_10, 8PSK_RM_24_5, 8PSK_RP_6, 8PSK_RP_12</td>
<td>QPSK_RM_32_10</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>TFCI</td>
<td>transport format combination indicator</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>TFCICode</td>
<td>coded TFCI</td>
<td>int</td>
</tr>
<tr>
<td>3</td>
<td>nTFCICodeWord</td>
<td>the number of bits in TFCI Code Word</td>
<td>int</td>
</tr>
</tbody>
</table>

**Notes/Equations**

1. This model encodes TFCI bits into TFCI code words.
Each firing, 1 nTFCICodeWord token and nTFCICodeWord TFCICode tokens are produced when 1 TFCI tokens are consumed.

Encoding of the TFCI depends on the modulation method and length. The relationship between the TFCICoding options, input length and the output length are described in Table 5-23.

<table>
<thead>
<tr>
<th>TFCICoding</th>
<th>Input bits</th>
<th>nTFCICodeWord</th>
<th>Description of TFCI Encoding</th>
</tr>
</thead>
<tbody>
<tr>
<td>QPSK_RP_4</td>
<td>1</td>
<td>4</td>
<td>If the number of TFCI bits is 1, then repetition will be used for coding. In this case each bit is repeated to a total of 4 times giving 4-bit transmission (NTFCI code word =4) for a single TFCI. For a single TFCI bit b0, the TFCI code word must be {b0, b0, b0, b0}.</td>
</tr>
<tr>
<td>QPSK_RP_8</td>
<td>2</td>
<td>8</td>
<td>If the number of TFCI bits is 2, then repetition will be used for coding. In this case each bit is repeated to a total of 4 times giving 8-bit transmission (NTFCI code word =8) for 2 TFCI bits. For two TFCI bits b0 and b1, the TFCI code word must be {b0, b1, b0, b1, b0, b1, b0, b1}.</td>
</tr>
<tr>
<td>QPSK_RM_16_5</td>
<td>3~5</td>
<td>16</td>
<td>If the number of TFCI bits is in the range 3~5, the TFCI is encoded using a (16, 5) bi-orthogonal (or first order Reed-Muller) code.</td>
</tr>
<tr>
<td>QPSK_RM_32_10</td>
<td>6~10</td>
<td>32</td>
<td>If the number of TFCI bits is in the range 6~10, the TFCI is encoded using a (32, 10) sub-code of the second order Reed-Muller code</td>
</tr>
<tr>
<td>_8PSK_RP_6</td>
<td>1</td>
<td>6</td>
<td>When the number of TFCI bits is 1, then repetition will be used for the coding. In this case, each bit is repeated to a total of 6 times giving 6-bit transmission (NTFCI code word = 6) for a single TFCI bit. For a single TFCI bit b0, the TFCI code word must be {b0, b0, b0, b0, b0, b0}.</td>
</tr>
<tr>
<td>_8PSK_RP_12</td>
<td>2</td>
<td>8</td>
<td>When the number of TFCI bits is 2, then repetition will be used for the coding. In this case, each bit is repeated to a total of 6 times giving 12-bit transmission (NTFCI code word = 12) for 2 TFCI bits. For two TFCI bits b0 and b1, the TFCI code word must be {b0, b1, b0, b1, b0, b1, b0, b1, b0, b1}.</td>
</tr>
<tr>
<td>_8PSK_RM_24_5</td>
<td>3~5</td>
<td>24</td>
<td>If the number of TFCI bits is in the range of 3 to 5, the TFCI bits are encoded using a (32,5) first order Reed-Muller code, then 8 bits out of 32 bits are punctured (Puncturing positions are 0, 1, 2, 3, 4, 5, 6, 7th bits).</td>
</tr>
<tr>
<td>_8PSK_RM_48_10</td>
<td>6~10</td>
<td>48</td>
<td>If the number of TFCI bits is in the range 6~10, the TFCI bits are encoded by using a (64,10) sub-code of the second order Reed-Muller code, then 16 bits out of 64 bits are punctured (Puncturing positions are 0, 4, 8, 13, 16, 20, 27, 31, 34, 38, 41, 44, 50, 54, 57, 61st bits).</td>
</tr>
</tbody>
</table>

References

Multiplexing and Coding Components

**TDSCDMA_TrChDeMux**

Description Transport channel demultiplexer  
Library TDSCDMA, Multiplexing & Coding  
Class SDFTDSCDMA_TrChDeMux  
Derived From TDSCDMA_CCTrCH_MuxBase  
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>PhyChNum_SA</td>
<td>physical channel allocation configuration</td>
<td>0 0 0 0 0 0</td>
<td>int array</td>
<td>[0, 2] for Uplink, [0, 16] for Downlink</td>
</tr>
<tr>
<td>TrChNum</td>
<td>number of Transport Channels</td>
<td>2</td>
<td>int</td>
<td>[1, 32]</td>
</tr>
<tr>
<td>RM_T A</td>
<td>rate matching attributes of all Transport Channels</td>
<td>1 1</td>
<td>int array</td>
<td>[1, 256]</td>
</tr>
<tr>
<td>DynTF_Set_T A</td>
<td>dynamic part of TF set of all Transport Channels</td>
<td>100 100 244 244</td>
<td>int array</td>
<td>[0, 5000] for transport block size, [0, 20000] for transport block set size</td>
</tr>
<tr>
<td>TF_SetSize_T A</td>
<td>transport format set size of all Transport Channels</td>
<td>1</td>
<td>int array</td>
<td>[1, 64] for each element</td>
</tr>
<tr>
<td>TTI_T A</td>
<td>transmission time interval of all Transport Channels</td>
<td>2 1</td>
<td>int array</td>
<td>[0, 3] for each element</td>
</tr>
<tr>
<td>CRC_T A</td>
<td>number of CRC bits of all Transport Channels</td>
<td>2 3</td>
<td>int array</td>
<td>[0, 4] for each element</td>
</tr>
<tr>
<td>ChCodingType_T A</td>
<td>channel coding type of all Transport Channels</td>
<td>2 2</td>
<td>int array</td>
<td>[0, 3] for each element</td>
</tr>
<tr>
<td>PuncLimit</td>
<td>puncturing limit</td>
<td>2/3</td>
<td>real</td>
<td>[0, 1]</td>
</tr>
<tr>
<td>Link</td>
<td>link selection: Downlink, Uplink</td>
<td>Uplink</td>
<td>enum</td>
<td></td>
</tr>
<tr>
<td>ModType_SA</td>
<td>type of modulation of all slots, 0 for QPSK, 1 for 8PSK</td>
<td>0 0 0 0 0 0 0</td>
<td>int array</td>
<td>[0, 1] for each element</td>
</tr>
</tbody>
</table>
This model is used to perform transport channel demultiplexing, the inverse operation of transport channel multiplexing. Each firing, 1 token consumed at TFCI indicates the value of transport format combination indicator; bm_PhyChNumAll \times \text{MAX\_BIT\_SLOT} tokens are consumed at DataIn, in which bm_PhyChNumAll is the number of allocated physical channels and MAX\_BIT\_SLOT is the possible maximum number of bits in one physical channel, that is 704 \times 3 \times 2; the maximum number of rate-matched tokens for all possible TFCI values in one frame for each transport channel are exported at multiple pin DataOut.

Each firing, one coded composite transport channel (CCTrCH) block is delivered to the transport channel demultiplexing. The CCTrCH block consists of frames.
Multiplexing and Coding Components

from each transport channel serially and this model demultiplexes these frames.

The bits input to the transport channel demultiplexing are denoted by $s_1, s_2, s_3, ..., s_S$, where $S$ is the number of bits in a CCTrCH block. The bits output are denoted by $f_{i1}, f_{i2}, f_{i3}, ..., f_{iV_i}$, where $i$ is the transport channel number and $V_i$ is the number of bits in the radio frame of transport channel $i$. The number of transport channels is denoted by $I$. $S = \sum_i V_i$. The transport channel demultiplexing is defined as follows.

$$S_k = f_{ik} \quad k = 1, 2, ..., V_1$$

$$S_k = f_{2.((V_i - 1))} \quad k = V_1 + 1, V_1 + 2, ..., V_1 + V_2$$

$$S_k = f_{3.((V_i + V_1))} \quad k = (V_1 + V_2) + 1, (V_1 + V_2) + 2, ..., (V_1 + V_2) + V_3$$

$$.$$ 

$$.$$ 

$$.$$ 

$$.$$ 

$$S_k = f_{.((V_i + V_2 + ... + V_{i-1}))} \quad k = (V_1 + V_2 + ... + V_{i-1}) + 1, (V_1 + V_2 + ... + V_{i-1}) + 2, ..., (V_1 + V_2 + ... + V_{i-1}) + V_i$$

3. All transport channel information must be provided in the form of arrays.

For DynTF_Set_TA the correct form is transport block size 1, transport block set size 1, transport block size 2, transport block set size 2, etc. The size of this array must be a multiple of 2, and the transport block set size must be a multiple of the relative transport block size.

When setting TTI_TA, CRC_TA and ChCodingType_TA, refer to Table 5-24.

| Column | Array Values
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>TTI_TA</td>
<td>Time</td>
</tr>
<tr>
<td>10ms</td>
<td>No CRC</td>
</tr>
<tr>
<td>20ms</td>
<td>1</td>
</tr>
<tr>
<td>40ms</td>
<td>2</td>
</tr>
<tr>
<td>80ms</td>
<td>3</td>
</tr>
</tbody>
</table>

5-80 TDSCDMA_TrChDeMux
4. PuncLimit denotes the variable PL defined in [2]. Refer to [2] for details regarding use of this variable in rate matching algorithm.

5. PhyChNum_SA indicates the number of allocated physical channels in each slot. The sum of PhyChNum_SA elements is the number of allocated physical channels.

6. TFCI_SA indicates in which slots TFCI bits will be transmitted. 0 denotes no TFCI bits will be transmitted in the slot. 1 denotes TFCI bits can be transmitted in the slot. The setting must be consistent with PhyChNum_SA setting, which means TFCI bits can only be transmitted in those slots in which the elements of PhyChNum_SA are not zero. Only the first allocated physical channel in each slot is used to transmit TFCI bits.

7. TFCI_Length_SA indicates the number of TFCI bits transmitted in each slot. If 0 is selected, the number of TFCI bits transmitted in the slot is dependent on the value of TFCI imported at TFCI pin. If a non-zero is selected, the number of TFCI bits transmitted in the slot is this non-zero value.

8. MinSF_PA indicates the minimum spread factor that can be used for corresponding physical channel. The size of MinSF_PA must be equal to the sum of PhyChNum_SA elements.

9. NdataOption specifies how the target number of rate-matched data is calculated with MinSF_PA. For Downlink, only Minimum can be selected. For Uplink, both can be selected. Refer to [2] for details.

10. SS_TPC_SA indicates the number of SS and TPC symbols transmitted in each slot. 1 denotes one SS and one TPC symbols are transmitted in the slot; 2 denotes no SS and no TPC symbols are transmitted in the slot; 3 denotes 16/SF SS and 16/SF TPC symbols are transmitted in the slot, where SF is the spreading factor of the physical channel used to transmitted SS and TPC symbols. Only the first allocated physical channel in each slot is used to transmit SS and TPC symbols.

References

Table 5-24. Array Values

<table>
<thead>
<tr>
<th>TTI_TA</th>
<th>CRC_TA</th>
<th>ChCodingType_TA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time</td>
<td>Value</td>
<td>Coding</td>
</tr>
<tr>
<td>24 bits</td>
<td>4</td>
<td></td>
</tr>
</tbody>
</table>

CC = convolutional coding; TC = turbo coding
Multiplexing and Coding Components

TDSCDMA_TrChMux

Description  Transport channel multiplexer
Library  TDSCDMA, Multiplexing & Coding
Class  SDFTDSCDMA_TrChMux
Derived From  TDSCDMA_CCTrCH_MuxBase
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>PhyChNum_SA</td>
<td>physical channel allocation configuration</td>
<td>0 0 1 0 0 0 0</td>
<td>int array</td>
<td>[0, 2] for Uplink, [0, 16] for Downlink</td>
</tr>
<tr>
<td>TrChNum</td>
<td>number of Transport Channels</td>
<td>2</td>
<td>int</td>
<td>[1, 32]</td>
</tr>
<tr>
<td>RM_TA</td>
<td>rate matching attributes of all Transport Channels</td>
<td>1 1</td>
<td>int array</td>
<td>[1, 256]</td>
</tr>
<tr>
<td>DynTF_Set_TA</td>
<td>dynamic part of TF set of all Transport Channels</td>
<td>100 100 244 244</td>
<td>int array</td>
<td>[0, 5000] for transport block size, [0, 20000] for transport block set size</td>
</tr>
<tr>
<td>TF_SetSize_TA</td>
<td>transport format set size of all Transport Channels</td>
<td>1 1</td>
<td>int array</td>
<td>[1, 64] for each element</td>
</tr>
<tr>
<td>TTI_TA</td>
<td>transmission time interval of all Transport Channels</td>
<td>2 1</td>
<td>int array</td>
<td>[0, 3] for each element</td>
</tr>
<tr>
<td>CRC_TA</td>
<td>number of CRC bits of all Transport Channels</td>
<td>2 3</td>
<td>int array</td>
<td>[0, 4] for each element</td>
</tr>
<tr>
<td>ChCodingType_TA</td>
<td>channel coding type of all Transport Channels</td>
<td>2 2</td>
<td>int array</td>
<td>[0, 3] for each element</td>
</tr>
<tr>
<td>PuncLimit</td>
<td>puncturing limit</td>
<td>2/3</td>
<td>real</td>
<td>[0, 1]</td>
</tr>
<tr>
<td>Link</td>
<td>link selection: Downlink, Uplink</td>
<td>Uplink</td>
<td>enum</td>
<td></td>
</tr>
<tr>
<td>ModType_SA</td>
<td>type of modulation of all slots, 0 for QPSK, 1 for 8PSK</td>
<td>0 0 0 0 0 0 0</td>
<td>int array</td>
<td>[0, 1] for each element</td>
</tr>
</tbody>
</table>

TDSCDMA_TrChMux  5-83
Multiplexing and Coding Components

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Default</th>
<th>Type</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>TFCl_SA</td>
<td>allocated TFCl transmitted</td>
<td>0 0 1 0 0 0 0</td>
<td>int array</td>
<td>[0, 1] for each element</td>
</tr>
<tr>
<td>TFCl_Length_SA</td>
<td>length of TFCl of all slots</td>
<td>0 0 1 0 0 0 0</td>
<td>int array</td>
<td>[0, 4, 8, 16, 32] for QPSK, [0, 6, 12, 24, 48] for 8PSK</td>
</tr>
<tr>
<td>SS_TPC_SA</td>
<td>type of SS and TPC of all slots</td>
<td>2 2 1 2 2 2 2 2</td>
<td>int array</td>
<td>[1, 3]</td>
</tr>
<tr>
<td>MinSF_PA</td>
<td>minimum spreading factor</td>
<td>8</td>
<td>int array</td>
<td>{1, 2, 4, 8, 16} for Uplink, {1, 16} for Downlink</td>
</tr>
<tr>
<td>NdataOption</td>
<td>the two options to determine the number Ndata: Minimum, Autonomous</td>
<td>Minimum</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>TFCl</td>
<td>transport channel format combination indicator</td>
<td>int</td>
</tr>
<tr>
<td>2</td>
<td>DataIn</td>
<td>input data</td>
<td>multiple 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>DataOut</td>
<td>output data</td>
<td>int</td>
</tr>
</tbody>
</table>

**Notes/Equations**

1. This model is used to perform transport channel multiplexing. Each firing, 1 token consumed at TFCl indicates the value of transport format combination indicator; the maximum number of rate-matched tokens for all possible TFCl values in one frame for each transport channel are consumed at multiple pin DataIn; bm_PhyChNumAll × MAX_BIT_SLOT tokens are exported at DataOut, in which bm_PhyChNumAll is the number of allocated physical channels and MAX_BIT_SLOT is the possible maximum number of bits in one physical channel, that is 704 × 3 × 2.

2. Every 10 ms, one radio frame from each transport channel is delivered to the transport channel multiplexing. These radio frames are serially multiplexed into a coded composite transport channel (CCTrCH).
The bits input to the transport channel multiplexing are denoted by $f_{i1}, f_{i2}, f_{i3}, ..., f_{iV_i}$, where $i$ is the transport channel number and $V_i$ is the number of bits in the radio frame of transport channel $i$. The number of transport channels is denoted by $I$. The bits output from transport channel multiplexing are denoted by $s_1, s_2, s_3, ..., s_S$, where $S$ is the number of bits, i.e. $S = \sum V_i$. The transport channel multiplexing is defined as follows.

$$S_k = f_{ik}, \quad k = 1, 2, ..., V_1$$
$$S_k = f_{2, (k-V_1)}, \quad k = V_1 + 1, V_1 + 2, ..., V_1 + V_2$$
$$S_k = f_{3, (V_1 + V_2)}, \quad k = (V_1 + V_2) + 1, (V_1 + V_2) + 2, ..., (V_1 + V_2) + V_3$$

... 

$$S_k = f_{I, (V_1 + V_2 + ... + V_{I-1})}, \quad k = (V_1 + V_2 + ... + V_{I-1}) + 1, (V_1 + V_2 + ... + V_{I-1}) + 2, ..., (V_1 + V_2 + ... + V_{I-1}) + V_I$$

3. All transport channel information must be provided in the form of arrays.

For DynTF_Set_TA the correct form is transport block size 1, transport block set size 1, transport block size 2, transport block set size 2, etc. The size of this array must be a multiple of 2, and the transport block set size must be a multiple of the relative transport block size.

When setting TTI_TA, CRC_TA and ChCodingType_TA, refer to Table 5-25.

<table>
<thead>
<tr>
<th>TTI_TA</th>
<th>CRC_TA</th>
<th>ChCodingType_TA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time</td>
<td>Value</td>
<td>Value</td>
</tr>
<tr>
<td>10ms</td>
<td>0</td>
<td>No CRC</td>
</tr>
<tr>
<td>20ms</td>
<td>1</td>
<td>8 bits</td>
</tr>
<tr>
<td>40ms</td>
<td>2</td>
<td>12 bits</td>
</tr>
<tr>
<td>80ms</td>
<td>3</td>
<td>16 bits</td>
</tr>
<tr>
<td></td>
<td>24 bits</td>
<td>4</td>
</tr>
</tbody>
</table>

CC = convolutional coding; TC = turbo coding
Multiplexing and Coding Components

4. PuncLimit denotes the variable PL defined in [2]. Refer to [2] for details regarding use of this variable in rate matching algorithm.

5. PhyChNum_SA indicates the number of allocated physical channels in each slot. The sum of PhyChNum_SA elements is the number of allocated physical channels.

6. TFCI_SA indicates in which slots TFCI bits will be transmitted. 0 denotes no TFCI bits will be transmitted in the slot. 1 denotes TFCI bits can be transmitted in the slot. The setting must be consistent with PhyChNum_SA setting, which means TFCI bits can only be transmitted in those slots in which the elements of PhyChNum_SA are not zero. Only the first allocated physical channel in each slot is used to transmit TFCI bits.

7. TFCI_Length_SA indicates the number of TFCI bits transmitted in each slot. If 0 is selected, the number of TFCI bits transmitted in the slot is dependent on the value of TFCI imported at TFCI pin. If a non-zero is selected, the number of TFCI bits transmitted in the slot is this non-zero value.

8. MinSF_PA indicates the minimum spread factor that can be used for corresponding physical channel. The size of MinSF_PA must be equal to the sum of PhyChNum_SA elements.

9. NdataOption specifies how the target number of rate-matched data is calculated with MinSF_PA. For Downlink, only Minimum can be selected. For Uplink, both can be selected. Refer to [2] for details.

10. SS_TPC_SA indicates the number of SS and TPC symbols transmitted in each slot. 1 denotes one SS and one TPC symbols are transmitted in the slot; 2 denotes no SS and no TPC symbols are transmitted in the slot; 3 denotes 16/SF SS and 16/SF TPC symbols are transmitted in the slot, where SF is the spreading factor of the physical channel used to transmitted SS and TPC symbols. Only the first allocated physical channel in each slot is used to transmit SS and TPC symbols.

References


Multiplexing and Coding Components
Chapter 6: Physical Channel Components
Physical Channel Components

TDSCDMA_DPCH

Description  Dedicated physical channel
Library  TDSCDMA, Physical Channels
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>SlotIndex</td>
<td>index of slot</td>
<td>2</td>
<td>int</td>
<td>[0, 6]</td>
</tr>
<tr>
<td>MidambleAllocScheme</td>
<td>midamble allocation scheme: UE_Specific, Common, Default</td>
<td>Default</td>
<td>enum</td>
<td></td>
</tr>
<tr>
<td>BasicMidambleID</td>
<td>index of basic midamble</td>
<td>0</td>
<td>int</td>
<td>[0, 127]</td>
</tr>
<tr>
<td>K</td>
<td>maximum number of midamble shifts in a cell</td>
<td>16</td>
<td>int</td>
<td>[2, 4, 6, 8, 10, 12, 14, 16]</td>
</tr>
<tr>
<td>MidambleID</td>
<td>index of midamble</td>
<td>1</td>
<td>int</td>
<td>[1, K]</td>
</tr>
<tr>
<td>SpreadFactor</td>
<td>spreading factor</td>
<td>16</td>
<td>int</td>
<td>(1, 2, 4, 8, 16) †</td>
</tr>
<tr>
<td>SpreadCode</td>
<td>index of spread code</td>
<td>1</td>
<td>int</td>
<td>[1, SpreadFactor]</td>
</tr>
<tr>
<td>PhyChNum</td>
<td>number of channelization codes used in a timeslot</td>
<td>1</td>
<td>int</td>
<td>[1, 16]</td>
</tr>
<tr>
<td>Link</td>
<td>link selection: Downlink, Uplink</td>
<td>Downlink</td>
<td>enum</td>
<td></td>
</tr>
<tr>
<td>ModType</td>
<td>type of modulation: QPSK, _8PSK</td>
<td>QPSK</td>
<td>enum</td>
<td></td>
</tr>
<tr>
<td>N_TFCI</td>
<td>number of TFCI bits</td>
<td>0</td>
<td>int</td>
<td>†</td>
</tr>
<tr>
<td>N_SS_N_TPC</td>
<td>number of SS and TPC</td>
<td>0</td>
<td>int</td>
<td>†</td>
</tr>
</tbody>
</table>

† N_TFCI, and N_SS_N_TPC values are given in Note 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 of dedicated physical channel</td>
<td>int</td>
</tr>
<tr>
<td>2</td>
<td>TFCI</td>
<td>transport format combination indicator bits</td>
<td>int</td>
</tr>
<tr>
<td>3</td>
<td>SS</td>
<td>information bits for uplink synchronization control</td>
<td>int</td>
</tr>
<tr>
<td>4</td>
<td>TPC</td>
<td>transmit power control 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>5</td>
<td>Output</td>
<td>data other than midamble in DPCH</td>
<td>complex</td>
</tr>
</tbody>
</table>

Notes/Equations

1. This subnetwork generates dedicated physical channel signals.

   The schematic for this subnetwork is shown in Figure 6-1.

   TDSCDMA_DPCH_DataMux multiplexes data, TFCI, SS and TPC bits. Data is then modulated and spread, then multiplexed with midamble and GP. The burst is placed in one subframe of slot according to the SlotIndex setting. Data other than slot will be all 0.

   ![Figure 6-1. TDSCDMA_DPCH Schematic](image)

2. Sub-frame and burst structure are illustrated in Figure 6-2 and Figure 6-3. Where Time slot #n (n = 0 to 6) is the nth traffic time slot, 864-chip duration;
Physical Channel Components

DwPTS is downlink pilot time slot, 96-chip duration; UpPTS is uplink pilot time slot, 160-chip duration; GP is main guard period for TDD operation, 96-chip duration.

Figure 6-2. Sub-Frame Structure for 1.28Mcps TDD Option

3. Time slot formats for the downlink with QPSK modulation are given in Table 6-1; time slot formats for uplink with QPSK modulation are given in Table 6-2; time slot formats for both links with 8PSK modulation are given in Table 6-3.

Table 6-1. Time Slot Formats for Downlink with QPSK Modulation

<table>
<thead>
<tr>
<th>Slot Format #</th>
<th>Spread Factor</th>
<th>NTFCI code word (bits)</th>
<th>NSS &amp; NTPC (bits)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>16</td>
<td>0</td>
<td>0 &amp; 0</td>
</tr>
<tr>
<td>1</td>
<td>16</td>
<td>4</td>
<td>0 &amp; 0</td>
</tr>
<tr>
<td>2</td>
<td>16</td>
<td>8</td>
<td>0 &amp; 0</td>
</tr>
<tr>
<td>3</td>
<td>16</td>
<td>16</td>
<td>0 &amp; 0</td>
</tr>
<tr>
<td>4</td>
<td>16</td>
<td>32</td>
<td>0 &amp; 0</td>
</tr>
<tr>
<td>5</td>
<td>16</td>
<td>0</td>
<td>2 &amp; 2</td>
</tr>
<tr>
<td>6</td>
<td>16</td>
<td>4</td>
<td>2 &amp; 2</td>
</tr>
<tr>
<td>7</td>
<td>16</td>
<td>8</td>
<td>2 &amp; 2</td>
</tr>
<tr>
<td>8</td>
<td>16</td>
<td>16</td>
<td>2 &amp; 2</td>
</tr>
</tbody>
</table>
### Table 6-1. Time Slot Formats for Downlink with QPSK Modulation

<table>
<thead>
<tr>
<th>Slot Format #</th>
<th>Spread Factor</th>
<th>NTFCI code word (bits)</th>
<th>NSS and NTPC (bits)</th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td>16</td>
<td>32</td>
<td>2 &amp; 2</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>0</td>
<td>0 &amp; 0</td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>4</td>
<td>0 &amp; 0</td>
</tr>
<tr>
<td>12</td>
<td>1</td>
<td>8</td>
<td>0 &amp; 0</td>
</tr>
<tr>
<td>13</td>
<td>1</td>
<td>16</td>
<td>0 &amp; 0</td>
</tr>
<tr>
<td>14</td>
<td>1</td>
<td>32</td>
<td>0 &amp; 0</td>
</tr>
<tr>
<td>15</td>
<td>1</td>
<td>0</td>
<td>2 &amp; 2</td>
</tr>
<tr>
<td>16</td>
<td>1</td>
<td>4</td>
<td>2 &amp; 2</td>
</tr>
<tr>
<td>17</td>
<td>1</td>
<td>8</td>
<td>2 &amp; 2</td>
</tr>
<tr>
<td>18</td>
<td>1</td>
<td>16</td>
<td>2 &amp; 2</td>
</tr>
<tr>
<td>19</td>
<td>1</td>
<td>32</td>
<td>2 &amp; 2</td>
</tr>
<tr>
<td>20</td>
<td>1</td>
<td>0</td>
<td>32 &amp; 32</td>
</tr>
<tr>
<td>21</td>
<td>1</td>
<td>4</td>
<td>32 &amp; 32</td>
</tr>
<tr>
<td>22</td>
<td>1</td>
<td>8</td>
<td>32 &amp; 32</td>
</tr>
<tr>
<td>23</td>
<td>1</td>
<td>16</td>
<td>32 &amp; 32</td>
</tr>
<tr>
<td>24</td>
<td>1</td>
<td>32</td>
<td>32 &amp; 32</td>
</tr>
</tbody>
</table>

### Table 6-2. Time Slot Formats for Uplink with QPSK Modulation

<table>
<thead>
<tr>
<th>Slot Format #</th>
<th>Spread Factor</th>
<th>NTFCI code word (bits)</th>
<th>NSS and NTPC (bits)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>16</td>
<td>0</td>
<td>0 &amp; 0</td>
</tr>
<tr>
<td>1</td>
<td>16</td>
<td>4</td>
<td>0 &amp; 0</td>
</tr>
<tr>
<td>2</td>
<td>16</td>
<td>8</td>
<td>0 &amp; 0</td>
</tr>
<tr>
<td>3</td>
<td>16</td>
<td>16</td>
<td>0 &amp; 0</td>
</tr>
<tr>
<td>4</td>
<td>16</td>
<td>32</td>
<td>0 &amp; 0</td>
</tr>
<tr>
<td>5</td>
<td>16</td>
<td>0</td>
<td>2 &amp; 2</td>
</tr>
<tr>
<td>6</td>
<td>16</td>
<td>4</td>
<td>2 &amp; 2</td>
</tr>
<tr>
<td>7</td>
<td>16</td>
<td>8</td>
<td>2 &amp; 2</td>
</tr>
<tr>
<td>8</td>
<td>16</td>
<td>16</td>
<td>2 &amp; 2</td>
</tr>
<tr>
<td>9</td>
<td>16</td>
<td>32</td>
<td>2 &amp; 2</td>
</tr>
<tr>
<td>10</td>
<td>8</td>
<td>0</td>
<td>0 &amp; 0</td>
</tr>
<tr>
<td>11</td>
<td>8</td>
<td>4</td>
<td>0 &amp; 0</td>
</tr>
<tr>
<td>12</td>
<td>8</td>
<td>8</td>
<td>0 &amp; 0</td>
</tr>
<tr>
<td>13</td>
<td>8</td>
<td>16</td>
<td>0 &amp; 0</td>
</tr>
</tbody>
</table>
Physical Channel Components

### Table 6-2. Time Slot Formats for Uplink with QPSK Modulation (continued)

<table>
<thead>
<tr>
<th>Slot Format #</th>
<th>Spread Factor</th>
<th>NTFCI code word (bits)</th>
<th>NSS and NTPC (bits)</th>
</tr>
</thead>
<tbody>
<tr>
<td>14</td>
<td>8</td>
<td>32</td>
<td>0 &amp; 0</td>
</tr>
<tr>
<td>15</td>
<td>8</td>
<td>0</td>
<td>2 &amp; 2</td>
</tr>
<tr>
<td>16</td>
<td>8</td>
<td>4</td>
<td>2 &amp; 2</td>
</tr>
<tr>
<td>17</td>
<td>8</td>
<td>8</td>
<td>2 &amp; 2</td>
</tr>
<tr>
<td>18</td>
<td>8</td>
<td>16</td>
<td>2 &amp; 2</td>
</tr>
<tr>
<td>19</td>
<td>8</td>
<td>32</td>
<td>2 &amp; 2</td>
</tr>
<tr>
<td>20</td>
<td>8</td>
<td>0</td>
<td>4 &amp; 4</td>
</tr>
<tr>
<td>21</td>
<td>8</td>
<td>4</td>
<td>4 &amp; 4</td>
</tr>
<tr>
<td>22</td>
<td>8</td>
<td>8</td>
<td>4 &amp; 4</td>
</tr>
<tr>
<td>23</td>
<td>8</td>
<td>16</td>
<td>4 &amp; 4</td>
</tr>
<tr>
<td>24</td>
<td>8</td>
<td>32</td>
<td>4 &amp; 4</td>
</tr>
<tr>
<td>25</td>
<td>4</td>
<td>0</td>
<td>0 &amp; 0</td>
</tr>
<tr>
<td>26</td>
<td>4</td>
<td>4</td>
<td>0 &amp; 0</td>
</tr>
<tr>
<td>27</td>
<td>4</td>
<td>8</td>
<td>0 &amp; 0</td>
</tr>
<tr>
<td>28</td>
<td>4</td>
<td>16</td>
<td>0 &amp; 0</td>
</tr>
<tr>
<td>29</td>
<td>4</td>
<td>32</td>
<td>0 &amp; 0</td>
</tr>
<tr>
<td>30</td>
<td>4</td>
<td>0</td>
<td>2 &amp; 2</td>
</tr>
<tr>
<td>31</td>
<td>4</td>
<td>4</td>
<td>2 &amp; 2</td>
</tr>
<tr>
<td>32</td>
<td>4</td>
<td>8</td>
<td>2 &amp; 2</td>
</tr>
<tr>
<td>33</td>
<td>4</td>
<td>16</td>
<td>2 &amp; 2</td>
</tr>
<tr>
<td>34</td>
<td>4</td>
<td>32</td>
<td>2 &amp; 2</td>
</tr>
<tr>
<td>35</td>
<td>4</td>
<td>0</td>
<td>8 &amp; 8</td>
</tr>
<tr>
<td>36</td>
<td>4</td>
<td>4</td>
<td>8 &amp; 8</td>
</tr>
<tr>
<td>37</td>
<td>4</td>
<td>8</td>
<td>8 &amp; 8</td>
</tr>
<tr>
<td>38</td>
<td>4</td>
<td>16</td>
<td>8 &amp; 8</td>
</tr>
<tr>
<td>39</td>
<td>4</td>
<td>32</td>
<td>8 &amp; 8</td>
</tr>
<tr>
<td>40</td>
<td>2</td>
<td>0</td>
<td>0 &amp; 0</td>
</tr>
<tr>
<td>41</td>
<td>2</td>
<td>4</td>
<td>0 &amp; 0</td>
</tr>
<tr>
<td>42</td>
<td>2</td>
<td>8</td>
<td>0 &amp; 0</td>
</tr>
<tr>
<td>43</td>
<td>2</td>
<td>16</td>
<td>0 &amp; 0</td>
</tr>
<tr>
<td>44</td>
<td>2</td>
<td>32</td>
<td>0 &amp; 0</td>
</tr>
<tr>
<td>45</td>
<td>2</td>
<td>0</td>
<td>2 &amp; 2</td>
</tr>
<tr>
<td>46</td>
<td>2</td>
<td>4</td>
<td>2 &amp; 2</td>
</tr>
<tr>
<td>47</td>
<td>2</td>
<td>8</td>
<td>2 &amp; 2</td>
</tr>
</tbody>
</table>
Table 6-2. Time Slot Formats for Uplink with QPSK Modulation (continued)

<table>
<thead>
<tr>
<th>Slot Format #</th>
<th>Spread Factor</th>
<th>NTFCI code word (bits)</th>
<th>NSS and NTPC (bits)</th>
</tr>
</thead>
<tbody>
<tr>
<td>48</td>
<td>2</td>
<td>16</td>
<td>2 &amp; 2</td>
</tr>
<tr>
<td>49</td>
<td>2</td>
<td>32</td>
<td>2 &amp; 2</td>
</tr>
<tr>
<td>50</td>
<td>2</td>
<td>0</td>
<td>16 &amp; 16</td>
</tr>
<tr>
<td>51</td>
<td>2</td>
<td>4</td>
<td>16 &amp; 16</td>
</tr>
<tr>
<td>52</td>
<td>2</td>
<td>8</td>
<td>16 &amp; 16</td>
</tr>
<tr>
<td>53</td>
<td>2</td>
<td>16</td>
<td>16 &amp; 16</td>
</tr>
<tr>
<td>54</td>
<td>2</td>
<td>32</td>
<td>16 &amp; 16</td>
</tr>
<tr>
<td>55</td>
<td>1</td>
<td>0</td>
<td>0 &amp; 0</td>
</tr>
<tr>
<td>56</td>
<td>1</td>
<td>4</td>
<td>0 &amp; 0</td>
</tr>
<tr>
<td>57</td>
<td>1</td>
<td>8</td>
<td>0 &amp; 0</td>
</tr>
<tr>
<td>58</td>
<td>1</td>
<td>16</td>
<td>0 &amp; 0</td>
</tr>
<tr>
<td>59</td>
<td>1</td>
<td>32</td>
<td>0 &amp; 0</td>
</tr>
<tr>
<td>60</td>
<td>1</td>
<td>0</td>
<td>2 &amp; 2</td>
</tr>
<tr>
<td>61</td>
<td>1</td>
<td>4</td>
<td>2 &amp; 2</td>
</tr>
<tr>
<td>62</td>
<td>1</td>
<td>8</td>
<td>2 &amp; 2</td>
</tr>
<tr>
<td>63</td>
<td>1</td>
<td>16</td>
<td>2 &amp; 2</td>
</tr>
<tr>
<td>64</td>
<td>1</td>
<td>32</td>
<td>2 &amp; 2</td>
</tr>
<tr>
<td>65</td>
<td>1</td>
<td>0</td>
<td>32 &amp; 32</td>
</tr>
<tr>
<td>66</td>
<td>1</td>
<td>4</td>
<td>32 &amp; 32</td>
</tr>
<tr>
<td>67</td>
<td>1</td>
<td>8</td>
<td>32 &amp; 32</td>
</tr>
<tr>
<td>68</td>
<td>1</td>
<td>16</td>
<td>32 &amp; 32</td>
</tr>
<tr>
<td>69</td>
<td>1</td>
<td>32</td>
<td>32 &amp; 32</td>
</tr>
</tbody>
</table>

Table 6-3. Time Slot Formats for 8PSK modulation

<table>
<thead>
<tr>
<th>Slot Format #</th>
<th>Spread Factor</th>
<th>NTFCI code word (bits)</th>
<th>NSS and NTPC (bits)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0 &amp; 0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>6</td>
<td>0 &amp; 0</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>12</td>
<td>0 &amp; 0</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>24</td>
<td>0 &amp; 0</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>48</td>
<td>0 &amp; 0</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>0</td>
<td>3 &amp; 3</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>6</td>
<td>3 &amp; 3</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>12</td>
<td>3 &amp; 3</td>
</tr>
</tbody>
</table>
Physical Channel Components

Table 6-3. Time Slot Formats for 8PSK modulation (continued)

<table>
<thead>
<tr>
<th>Slot Format #</th>
<th>Spread Factor</th>
<th>NTFCI code word (bits)</th>
<th>NSS and NTPC (bits)</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>1</td>
<td>24</td>
<td>3 &amp; 3</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>48</td>
<td>3 &amp; 3</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>0</td>
<td>48 &amp; 48</td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>6</td>
<td>48 &amp; 48</td>
</tr>
<tr>
<td>12</td>
<td>1</td>
<td>12</td>
<td>48 &amp; 48</td>
</tr>
<tr>
<td>13</td>
<td>1</td>
<td>24</td>
<td>48 &amp; 48</td>
</tr>
<tr>
<td>14</td>
<td>1</td>
<td>48</td>
<td>48 &amp; 48</td>
</tr>
<tr>
<td>15</td>
<td>16</td>
<td>0</td>
<td>0 &amp; 0</td>
</tr>
<tr>
<td>16</td>
<td>16</td>
<td>6</td>
<td>0 &amp; 0</td>
</tr>
<tr>
<td>17</td>
<td>16</td>
<td>12</td>
<td>0 &amp; 0</td>
</tr>
<tr>
<td>18</td>
<td>16</td>
<td>24</td>
<td>0 &amp; 0</td>
</tr>
<tr>
<td>19</td>
<td>16</td>
<td>48</td>
<td>0 &amp; 0</td>
</tr>
<tr>
<td>20</td>
<td>16</td>
<td>0</td>
<td>3 &amp; 3</td>
</tr>
<tr>
<td>21</td>
<td>16</td>
<td>6</td>
<td>3 &amp; 3</td>
</tr>
<tr>
<td>22</td>
<td>16</td>
<td>12</td>
<td>3 &amp; 3</td>
</tr>
<tr>
<td>23</td>
<td>16</td>
<td>24</td>
<td>3 &amp; 3</td>
</tr>
<tr>
<td>24</td>
<td>16</td>
<td>48</td>
<td>3 &amp; 3</td>
</tr>
</tbody>
</table>

References

TDSCDMA_DwPCH

Description  Downlink synchronization channel generation
Library   TDSCDMA, Physical Channels
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>SyncCode</td>
<td>index of basic synchronization code</td>
<td>0</td>
<td>int</td>
<td>[0, 31]</td>
</tr>
<tr>
<td>ModPhase</td>
<td>type of modulation quadruples: S1, S2</td>
<td>S1</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>output</td>
<td>downlink sync code output</td>
<td>complex</td>
</tr>
</tbody>
</table>

Notes/Equations

1. This model generates downlink synchronization channel signals.

   The schematic for this subnetwork is shown in Figure 6-4.

   Each firing, one sub-frame containing 6400 chips is exported. The downlink synchronization sequence containing 64 chips is exported at location 897 ~ 960. Data at other locations are 0s.

2. Four consecutive phases of the downlink synchronization sequence are used to indicate the presence of the P-CCPCH in the next 4 sub-frames. If ModPhase=S1, there is a P-CCPCH in the next 4 sub-frames; if ModPhase=S2, there is no P-CCPCH in the next 4 sub-frames.
Physical Channel Components

Figure 6-4. TDSCDMA_DwPCH Schematic
References

Physical Channel Components

**TDSCDMA_FPACH**

**Description**  
Fast physical access channel

**Library**  
TDSCDMA, Physical Channels

**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>MidambleAllocScheme</td>
<td>midamble allocation scheme: UE_Specific, Common, Default</td>
<td>Default</td>
<td>enum</td>
<td></td>
</tr>
<tr>
<td>BasicMidambleID</td>
<td>index of basic midamble</td>
<td>0</td>
<td>int</td>
<td>[0, 127]</td>
</tr>
<tr>
<td>K</td>
<td>maximum number of midamble shifts in a cell</td>
<td>16</td>
<td>int</td>
<td>(2, 4, 6, 8, 10, 12, 14, 16)</td>
</tr>
<tr>
<td>MidambleID</td>
<td>index of midamble</td>
<td>1</td>
<td>int</td>
<td>[1, K]</td>
</tr>
<tr>
<td>SpreadCode</td>
<td>index of spread code</td>
<td>2</td>
<td>int</td>
<td>[1, 16]</td>
</tr>
<tr>
<td>PhyChNum</td>
<td>number of channelization codes used in a timeslot</td>
<td>1</td>
<td>int</td>
<td>[1, 16]</td>
</tr>
<tr>
<td>ModType</td>
<td>type of modulation: QPSK, _8PSK</td>
<td>QPSK</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 for FPACH</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>output data</td>
<td>complex</td>
</tr>
</tbody>
</table>

**Notes/Equations**

6-12  
TDSCDMA_FPACH
1. This subnetwork generates fast physical access channel signals. The FPACH is used by Node B to carry, in a single burst, the detected signature with timing and power level adjustment indicator to the user equipment. FPACH uses one code with spreading factor of 16, so that its burst is composed of 44 symbols. The spreading code, training sequence, and time slot position are configured by the network and signalled on the BCH.

The schematic for this subnetwork is shown in Figure 6-5.

References

Physical Channel Components

**TDSCDMA_PCCPCH**

Description  Primary common control channel
Library   TDSCDMA, Physical Channels
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>BasicMidambleID</td>
<td>index of basic midamble</td>
<td>0</td>
<td>int</td>
<td>[0, 127]</td>
</tr>
<tr>
<td>SCTD</td>
<td>Space code transmit diversity flag: ON, OFF</td>
<td>OFF</td>
<td>enum</td>
<td></td>
</tr>
<tr>
<td>ModType</td>
<td>type of modulation: QPSK, 8PSK</td>
<td>QPSK</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>Input1</td>
<td>Input data for P-CCPCH1</td>
<td>int</td>
</tr>
<tr>
<td>2</td>
<td>Input2</td>
<td>Input data for P-CCPCH2</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>output data</td>
<td>complex</td>
</tr>
</tbody>
</table>

### Notes/Equations

1. This subnetwork generates primary common control physical channel signals. The schematic for this subnetwork is shown in Figure 6-6.

2. P-CCPCHs are mapped onto the first two code channels of timeslot 0. The P-CCPCH is always transmitted with an antenna pattern configuration that provides whole cell coverage.
The P-CCPCH uses fixed spreading with a spreading factor SF = 16. P-CCPCH1 and P-CCPCHP2 always use first and second channelization codes, respectively. The training sequences (midambles) are used for the P-CCPCH. For timeslot 0 in which the P-CCPCH is transmitted, midambles $m_1^{(1)}$ and $m_2^{(2)}$ are reserved for P-CCPCH in order to support Space Code Transmit Diversity (SCTD) and the beacon function. The use of midambles depends on whether SCTD is applied to the P-CCPCH. If antenna diversity is not applied to P-CCPCH, $m_1^{(1)}$ is used and $m_2^{(2)}$ is left unused. Otherwise, $m_1^{(1)}$ is used for the first antenna and $m_2^{(2)}$ is used for the diversity antenna.

Figure 6-6. TDSCDMA_PCCPCH Schematic

References

Physical Channel Components

TDSCDMA_PICH

Description  Page indicator channel
Library  TDSCDMA, Physical Channels
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>SlotIndex</td>
<td>index of slot</td>
<td>6</td>
<td>int</td>
<td>(0, 2, 3, 4, 5, 6)</td>
</tr>
<tr>
<td>MidambleAllocScheme</td>
<td>midamble allocation scheme: UE_Specific, Common, Default</td>
<td>Default</td>
<td>enum</td>
<td></td>
</tr>
<tr>
<td>BasicMidambleID</td>
<td>index of basic midamble</td>
<td>0</td>
<td>int</td>
<td>[0, 127]</td>
</tr>
<tr>
<td>K</td>
<td>maximum number of midamble shifts in a cell</td>
<td>16</td>
<td>int</td>
<td>(2, 4, 6, 8, 10, 12, 14, 16)</td>
</tr>
<tr>
<td>MidambleID1</td>
<td>index of midamble for S-CCPCH1</td>
<td>5</td>
<td>int</td>
<td>[1, K]</td>
</tr>
<tr>
<td>MidambleID2</td>
<td>index of midamble for S-CCPCH2</td>
<td>6</td>
<td>int</td>
<td>[1, K]</td>
</tr>
<tr>
<td>SpreadCode1</td>
<td>index of spread code for S-CCPCH1</td>
<td>5</td>
<td>int</td>
<td>[1, 16]</td>
</tr>
<tr>
<td>SpreadCode2</td>
<td>index of spread code for S-CCPCH2</td>
<td>6</td>
<td>int</td>
<td>[1, 16]</td>
</tr>
<tr>
<td>PhyChNum</td>
<td>number of channelization codes used in a timeslot</td>
<td>2</td>
<td>int</td>
<td>[1, 16]</td>
</tr>
<tr>
<td>ModType</td>
<td>type of modulation: QPSK, 8PSK</td>
<td>QPSK</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</td>
<td>int</td>
</tr>
</tbody>
</table>

6-16  TDSCDMA_PICH
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 data</td>
<td>complex</td>
</tr>
</tbody>
</table>

Notes/Equations

1. This subnetwork generates page indicator channel signals.

The schematic for this subnetwork is shown in Figure 6-7 and Figure 6-8.

Figure 6-7. TDSCDMA_PICH Schematic (1 of 2)
2. Figure 6-9 illustrates the structure of a PICH transmission and the numbering of bits within the bursts. $N_{PIB}$ bits are used to carry the paging indicators, where $N_{PIB} = 352$. The PICH uses fixed spreading with a spreading factor $SF = 16$. 

Figure 6-8. TDSCDMA_PICH Schematic (2 of 2)
Figure 6-9. Transmission of Paging Indicator Carrying Bits in PICH Bursts

References

Physical Channel Components

TDSCDMA_PRACH

Description  Physical random access channel
Library  TDSCDMA, Physical Channels
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>SlotIndex</td>
<td>index of slot</td>
<td>1</td>
<td>int</td>
<td>[0, 6]</td>
</tr>
<tr>
<td>MidambleAllocScheme</td>
<td>midamble allocation scheme: UE_Specific, Common, Default</td>
<td>Default</td>
<td>enum</td>
<td></td>
</tr>
<tr>
<td>BasicMidambleID</td>
<td>index of basic midamble</td>
<td>0</td>
<td>int</td>
<td>[0, 127]</td>
</tr>
<tr>
<td>K</td>
<td>maximum number of midamble shifts in a cell</td>
<td>16</td>
<td>int</td>
<td>{2, 4, 6, 8, 10, 12, 14, 16}</td>
</tr>
<tr>
<td>MidambleID</td>
<td>index of midamble</td>
<td>1</td>
<td>int</td>
<td>[1, K]</td>
</tr>
<tr>
<td>SpreadFactor</td>
<td>spreading factor</td>
<td>16</td>
<td>int</td>
<td>{4, 8, 16}</td>
</tr>
<tr>
<td>SpreadCode</td>
<td>index of spread code</td>
<td>1</td>
<td>int</td>
<td>{1, SpreadFactor}</td>
</tr>
<tr>
<td>PhyChNum</td>
<td>number of channelization codes used in a timeslot</td>
<td>1</td>
<td>int</td>
<td>[1, 16]</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 of physical random access channel</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>data other than midamble in PRACH</td>
<td>complex</td>
</tr>
</tbody>
</table>
Notes/Equations

1. This subnetwork generates physical random access channel data.

   Each firing, for QPSK, 864 Output tokens are produced when 1408/SpreadFactor Data tokens are consumed. For 8PSK, 864 Output tokens are produced when 2112/SpreadFactor Data tokens are consumed.

   The schematic for this subnetwork is shown in Figure 6-10.

![Figure 6-10. TDSCDMA_PRACH Schematic](image)

References

Physical Channel Components

**TDSCDMA_PSCH**

**Description**  Uplink/downlink physical shared channel
**Library**  TDSCDMA, Physical Channels
**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>SlotIndex</td>
<td>index of slot</td>
<td>2</td>
<td>int</td>
<td>[0, 6]</td>
</tr>
<tr>
<td>MidambleAllocScheme</td>
<td>midamble allocation scheme: UE_Specific, Common, Default</td>
<td>Default</td>
<td>enum</td>
<td></td>
</tr>
<tr>
<td>BasicMidambleID</td>
<td>index of basic midamble</td>
<td>0</td>
<td>int</td>
<td>[0, 127]</td>
</tr>
<tr>
<td>K</td>
<td>maximum number of midamble shifts in a cell</td>
<td>16</td>
<td>int</td>
<td>[2, 4, 6, 8, 10, 12, 14, 16]</td>
</tr>
<tr>
<td>MidambleID</td>
<td>index of midamble</td>
<td>1</td>
<td>int</td>
<td>[1, K]</td>
</tr>
<tr>
<td>SpreadFactor</td>
<td>spreading factor</td>
<td>16</td>
<td>int</td>
<td>[1, SpreadFactor]</td>
</tr>
<tr>
<td>SpreadCode</td>
<td>index of spreading code</td>
<td>1</td>
<td>int</td>
<td>[1, SpreadFactor]</td>
</tr>
<tr>
<td>PhyChNum</td>
<td>number of channelization codes used in a timeslot</td>
<td>1</td>
<td>int</td>
<td>[1, 16]</td>
</tr>
<tr>
<td>Link</td>
<td>link selection: Downlink, Uplink</td>
<td>Downlink</td>
<td>enum</td>
<td></td>
</tr>
<tr>
<td>ModType</td>
<td>type of modulation: QPSK, BPSK</td>
<td>QPSK</td>
<td>enum</td>
<td></td>
</tr>
<tr>
<td>N_TFCI</td>
<td>number of TFCI bits</td>
<td>0</td>
<td>int</td>
<td>[0, 4, 8, 16, 32]      for QPSK; [0, 6, 12, 24, 48] for BPSK</td>
</tr>
</tbody>
</table>
1. This subnetwork generates physical downlink/uplink shared channel data. The schematic for this subnetwork is shown in Figure 6-11.

Each firing:

- for QPSK, 864 Output tokens are produced when \((1408/\text{SpreadFactor}-N_{\text{TFCI}}/2)\) Data tokens and \(N_{\text{TFCI}}/2\) TFCI tokens are consumed;

- for 8PSK, 864 Output tokens are produced when \((2112/\text{SpreadFactor}-N_{\text{TFCI}}/2)\) Data tokens and \(N_{\text{TFCI}}/2\) TFCI tokens are consumed.

Physical downlink/uplink shared channel provides for transmission of TFCI.

### 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 of physical shared channel</td>
<td>int</td>
</tr>
<tr>
<td>2</td>
<td>TFCI</td>
<td>transport format combination indicator 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>3</td>
<td>Output</td>
<td>data other than midamble in PSCH</td>
<td>complex</td>
</tr>
</tbody>
</table>

### Notes/Equations

- Each firing:
  - for QPSK, 864 Output tokens are produced when \((1408/\text{SpreadFactor}-N_{\text{TFCI}}/2)\) Data tokens and \(N_{\text{TFCI}}/2\) TFCI tokens are consumed;
  - for 8PSK, 864 Output tokens are produced when \((2112/\text{SpreadFactor}-N_{\text{TFCI}}/2)\) Data tokens and \(N_{\text{TFCI}}/2\) TFCI tokens are consumed.

Physical downlink/uplink shared channel provides for transmission of TFCI.

Figure 6-11. TDSCDMA_PSCH Schematic
Physical Channel Components

References

TDSCDMA_SCCPCH

Description  Secondary common control channel
Library  TDSCDMA, Physical Channels
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>SlotIndex</td>
<td>index of slot</td>
<td>6</td>
<td>int</td>
<td>{0, 2,3,4,5,6}</td>
</tr>
<tr>
<td>MidambleAllocScheme</td>
<td>midamble allocation scheme: UE_Specific, Common, Default</td>
<td>Default</td>
<td>enum</td>
<td></td>
</tr>
<tr>
<td>BasicMidambleID</td>
<td>index of basic midamble</td>
<td>0</td>
<td>int</td>
<td>[0, 127]</td>
</tr>
<tr>
<td>K</td>
<td>maximum number of midamble shifts in a cell</td>
<td>16</td>
<td>int</td>
<td>{2, 4,6,8,10,12,14,16}</td>
</tr>
<tr>
<td>MidambleID1</td>
<td>index of midamble for S-CCPCH1</td>
<td>2</td>
<td>int</td>
<td>[1, K]</td>
</tr>
<tr>
<td>MidambleID2</td>
<td>index of midamble for S-CCPCH2</td>
<td>3</td>
<td>int</td>
<td>[1, K]</td>
</tr>
<tr>
<td>SpreadCode1</td>
<td>index of spread code for S-CCPCH1</td>
<td>2</td>
<td>int</td>
<td>[1, 16]</td>
</tr>
<tr>
<td>SpreadCode2</td>
<td>index of spread code for S-CCPCH2</td>
<td>3</td>
<td>int</td>
<td>[1, 16]</td>
</tr>
<tr>
<td>PhyChNum</td>
<td>number of channelization codes used in a timeslot</td>
<td>2</td>
<td>int</td>
<td>[1, 16]</td>
</tr>
<tr>
<td>ModType</td>
<td>type of modulation: QPSK, 8PSK</td>
<td>QPSK</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>Input1</td>
<td>Input data for S-CCPCH1</td>
<td>int</td>
</tr>
<tr>
<td>2</td>
<td>Input2</td>
<td>Input data for S-CCPCH2</td>
<td>int</td>
</tr>
</tbody>
</table>
Physical Channel 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 data</td>
<td>complex</td>
</tr>
</tbody>
</table>

Notes/Equations

1. This subnetwork generates secondary common control physical channels. S-CCPCH 1 and S-CCPCH 2 are always used in pairs, mapped onto two code channels with a spreading factor of 16. There can be more than one pair of S-CCPCHs in use in one cell.

The schematic for this subnetwork is shown in Figure 6-12.

![Figure 6-12. TDSCDMA_SCCPCH Schematic](image)

References

TDSCDMA_UpPCH

Description  Uplink synchronization channel generation
Library    TDSCDMA, Physical Channels
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>SyncCode</td>
<td>index of basic synchronization code</td>
<td>0</td>
<td>int</td>
<td>[0, 255]</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>output</td>
<td>uplink sync code</td>
<td>complex</td>
</tr>
</tbody>
</table>

Notes/Equations

1. This subnetwork generates an uplink synchronization channel.
   The schematic for this subnetwork is shown in Figure 6-13.
   Each firing, one sub-frame containing 6400 chips is exported. The uplink synchronization sequence containing 128 chips is exported at location 1057 ~ 1184. Data at other locations is 0.
Physical Channel Components

![Figure 6-13. TDSCDMA_UpPCH Schematic](image)

References

Chapter 7: Receivers
Receivers

**TDSCDMA_12_2_DL_JD_Receiver**

Description  
Downlink joint detection receiver with 8 DPCH0

Library  
TDSCDMA, Receiver

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>FCarrier</td>
<td>carrier frequency</td>
<td>1900MHz</td>
<td>Hz</td>
<td>real</td>
<td>(-1) or (0, ∞)†</td>
</tr>
<tr>
<td>SamplesPerSymbol</td>
<td>samples per symbol</td>
<td>8</td>
<td>int</td>
<td></td>
<td>[1, 32]</td>
</tr>
<tr>
<td>SlotIndex</td>
<td>slot index</td>
<td>6</td>
<td>int</td>
<td></td>
<td>[1, 6]</td>
</tr>
<tr>
<td>BasicMidambleID</td>
<td>index of basic midamble</td>
<td>0</td>
<td>int</td>
<td></td>
<td>[0, 127]</td>
</tr>
<tr>
<td>K</td>
<td>maximum number of midamble shifts in a cell</td>
<td>16</td>
<td>int</td>
<td></td>
<td>[2, 4, 6, 8, 10, 12, 14, 16]</td>
</tr>
<tr>
<td>MidambleID</td>
<td>midamble index</td>
<td>1</td>
<td>int</td>
<td></td>
<td>[1, K]</td>
</tr>
<tr>
<td>OCNS_MidambleID_PA</td>
<td>OCNS midamble array</td>
<td>{2, 3, 4, 5, 6, 7, 8, 9}</td>
<td>int array</td>
<td>[1, K]</td>
<td></td>
</tr>
<tr>
<td>SpreadCode1</td>
<td>spreading code for the first DPCH</td>
<td>1</td>
<td>int</td>
<td></td>
<td>[1, 16]</td>
</tr>
<tr>
<td>SpreadCode2</td>
<td>spreading code for the second DPCH</td>
<td>2</td>
<td>int</td>
<td></td>
<td>[1, 16]</td>
</tr>
<tr>
<td>OCNS_SpreadCode_PA</td>
<td>OCNS spreading code array</td>
<td>{3, 4, 5, 6, 7, 8, 9, 10}</td>
<td>int array</td>
<td>[1, 16]</td>
<td></td>
</tr>
<tr>
<td>FilterLength</td>
<td>length of raised cosine filters in number of symbols</td>
<td>16</td>
<td>int</td>
<td></td>
<td>[1, ∞)</td>
</tr>
<tr>
<td>SystemDelay</td>
<td>total system delay in symbols including delay caused by filters</td>
<td>16</td>
<td>int</td>
<td></td>
<td>[1, ∞)</td>
</tr>
<tr>
<td>IgnoreNumber</td>
<td>ignored subframe numbers</td>
<td>4</td>
<td>int</td>
<td></td>
<td>[1, ∞)</td>
</tr>
<tr>
<td>PowerThreshold</td>
<td>power threshold for channel estimation</td>
<td>0</td>
<td>real</td>
<td></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>InRF</td>
<td>input signals</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>output</td>
<td>output</td>
<td>int</td>
</tr>
</tbody>
</table>

Notes/Equations

1. This subnetwork is used to implement a 12.2k DL JD receiver.

The schematic for this subnetwork is shown in Figure 7-1.
The ZF-BLE (zero forcing block linear equalization) joint detection algorithm is applied in this model.

Let

\[ d^{(k)} = (d_1^{(k)}, d_2^{(k)}, \ldots, d_N^{(k)})^T, \quad k = 1, \ldots, K \]

where \( K \) is the number of users, \( N \) is the number of information bits. \( A \) is the structure matrix, which is defined in document for TDSCDMA_A_Generator. And \( n \) is the stationary white Gaussian noise. Then the received sequence \( e \) can be written as

\[ e = (e_1, e_2, \ldots, e_N \times Q + W^{-1})^T = A d + n \]

where \( W \) is the length of channel impulse response.

In ZF-BLE algorithm, the estimate \( \hat{d} \) can be obtained by optimizing
where $R_{n}^{-1}$ is the noise covariance matrix.

Suppose $R_{n}^{-1} = I$, then from the estimation theory,

$$
\hat{d} = (A^T \times A)^{-1} \times A^T \times e.
$$

References


**TDSCDMA_12_2_DL_RakeReceiver**

**Description**  
Downlink rake receiver

**Library**  
TDSCDMA, Receiver

**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>FCarrier</td>
<td>carrier frequency</td>
<td>1900MHz</td>
<td>Hz</td>
<td>real</td>
<td>[-1] or (0, ∞)†</td>
</tr>
<tr>
<td>AWGN</td>
<td>AWGN channel or not: No, Yes</td>
<td>Yes</td>
<td>enum</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SamplesPerSymbol</td>
<td>samples per symbol</td>
<td>8</td>
<td>int</td>
<td>[1, 32]</td>
<td></td>
</tr>
<tr>
<td>SlotIndex</td>
<td>slot index</td>
<td>6</td>
<td>int</td>
<td>[1, 6]</td>
<td></td>
</tr>
<tr>
<td>BasicMidambleID</td>
<td>index of basic midamble</td>
<td>0</td>
<td>int</td>
<td>[0, 127]</td>
<td></td>
</tr>
<tr>
<td>K</td>
<td>maximum number of midamble shifts in a cell</td>
<td>16</td>
<td>int</td>
<td>[2, 4, 6, 8, 10, 12, 14, 16]</td>
<td></td>
</tr>
<tr>
<td>MidambleID</td>
<td>midamble index</td>
<td>1</td>
<td>int</td>
<td>[1, K]</td>
<td></td>
</tr>
<tr>
<td>SpreadCode1</td>
<td>spreading code for the first DPCCH</td>
<td>1</td>
<td>int</td>
<td>[1, 16]</td>
<td></td>
</tr>
<tr>
<td>SpreadCode2</td>
<td>spreading code for the second DPCCH</td>
<td>2</td>
<td>int</td>
<td>[1, 16]</td>
<td></td>
</tr>
<tr>
<td>FilterLength</td>
<td>length of raised cosine filters in number of symbols</td>
<td>16</td>
<td>int</td>
<td>[1, ∞]</td>
<td></td>
</tr>
<tr>
<td>SystemDelay</td>
<td>total system delay in symbols including delay caused by filters</td>
<td>16</td>
<td>int</td>
<td>[1, ∞]</td>
<td></td>
</tr>
<tr>
<td>IgnoreNumber</td>
<td>ignored subframe numbers</td>
<td>4</td>
<td>int</td>
<td>[1, ∞]</td>
<td></td>
</tr>
<tr>
<td>PowerThreshold</td>
<td>power threshold for channel estimation</td>
<td>0</td>
<td>real</td>
<td>(0, ∞)</td>
<td></td>
</tr>
<tr>
<td>PhyChNum_SA</td>
<td>physical channel allocation configuration</td>
<td>0 0 2 0 0 0 0 0</td>
<td>int array</td>
<td>[0, 16] for Downlink, [0, 2] for Uplink</td>
<td></td>
</tr>
</tbody>
</table>

† If AWGN is set to "No", FCarrier is set to 1900MHz.
This subnetwork is used to implement 12.2k DL Rake receiver. The schematic for this subnetwork is shown in Figure 7-2.

<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>MaxPhyChNum</td>
<td>sum of allocated physical channel in all slots</td>
<td>2</td>
<td>int</td>
<td>[1, 112]</td>
<td></td>
</tr>
<tr>
<td>ModType_SA</td>
<td>type of modulation of all slots, 0 for QPSK, 1 for 8PSK</td>
<td>0 0 0 0 0 0</td>
<td>int array</td>
<td>[0, 1)</td>
<td></td>
</tr>
<tr>
<td>TFCI_SA</td>
<td>allocated TFCI transmitted active slots configuration</td>
<td>0 0 1 0 0 0 0</td>
<td>int array</td>
<td>[0, 1)</td>
<td></td>
</tr>
<tr>
<td>TFCI_Length_SA</td>
<td>length of TFCI of all slots</td>
<td>0 0 1 6 0 0 0 0</td>
<td>int array</td>
<td>[0, 4, 8, 16, 32) for QPSK, (0,6,12,24,48) for 8PSK</td>
<td></td>
</tr>
<tr>
<td>SS_TPC_SA</td>
<td>type of SS and TPC of all slots</td>
<td>2 2 1 2 2 2 2</td>
<td>int array</td>
<td>[1, 2, 3)</td>
<td></td>
</tr>
<tr>
<td>MinSF_PA</td>
<td>minimum spreading factor array corresponding to allocated physical channels</td>
<td>16 16</td>
<td>int array</td>
<td>[1, 16) for Downlink (1,2,4,8,16) for Uplink</td>
<td></td>
</tr>
</tbody>
</table>

† The FCarrier parameter sets the internal oscillator frequency used for demodulation. Setting FCarrier to -1 will use the input signal characterization frequency as the internal oscillator frequency.

**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>InRF</td>
<td>input signals</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>output</td>
<td>output</td>
<td>int</td>
</tr>
</tbody>
</table>
2. The received signal is demodulated to baseband and passed through a root raised-cosine filter. Certain delay is added to handle the mis-alignment caused by the filters in the transmitter and the receiver. The baseband signal is then demultiplexed and separated into two parts: midamble signal and data signal. The midamble signal is used to estimate the channel impulse response and further construct Matrix A. The Matrix A and data signal are used to estimate the original data symbols.

Different algorithms, RAKE or Joint Detection (JD), can be used in the estimator. In this subnetwork, the core algorithm is RAKE (also called discrete matched filter). The interference caused by multiple users is ignored. The optimal criteria is to maximize the SNR at the output. If the ZF-BLE JD algorithm is applied, which is the zero forcing block linear equalization algorithm, the noise is ignored, while the interference caused by multiple users is totally eliminated.

After a QPSK/8PSK demodulation, data symbol is converted to data bits. The TFCI, SS and TPC bits are dropped while the information bits are decoded.
Please refer to TDSCDMA_ChannelEstimation, TDSCDMA_A_Generator, TDSCDMA_b_k_Generator, TDSCDMA_RAKE and TDSCDMA_JointDetection for detail information on channel estimation and RAKE/JD core algorithm.

References


Receivers

**TDSCDMA_12_2_UL_JD_Receiver**

**Description**  Uplink joint detection receiver with 4 DPCH0

**Library**  TDSCDMA, Receiver

**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>FCarrier</td>
<td>carrier frequency</td>
<td>1900MHz</td>
<td>Hz</td>
<td>real</td>
<td>(-1) or (0, ∞)†</td>
</tr>
<tr>
<td>SamplesPerSymbol</td>
<td>samples per symbol</td>
<td>8</td>
<td>int</td>
<td></td>
<td>[1, 32]</td>
</tr>
<tr>
<td>SlotIndex</td>
<td>slot index</td>
<td>2</td>
<td>int</td>
<td></td>
<td>[1, 6]</td>
</tr>
<tr>
<td>BasicMidambleID</td>
<td>index of basic midamble</td>
<td>0</td>
<td>int</td>
<td></td>
<td>[0, 127]</td>
</tr>
<tr>
<td>K</td>
<td>maximum number of midamble shifts in a cell</td>
<td>16</td>
<td>int</td>
<td></td>
<td>{2, 4, 6, 8, 10, 12, 14, 16}</td>
</tr>
<tr>
<td>MidambleID</td>
<td>midamble index</td>
<td>1</td>
<td>int</td>
<td></td>
<td>[1, K]</td>
</tr>
<tr>
<td>OCNS_MidambleID_PA</td>
<td>OCNS midamble array</td>
<td>{2, 3, 4, 5}</td>
<td>int array</td>
<td>[1, K]</td>
<td></td>
</tr>
<tr>
<td>SpreadCode</td>
<td>spreading code for the first DPCH</td>
<td>1</td>
<td>int</td>
<td></td>
<td>[1, 16]</td>
</tr>
<tr>
<td>OCNS_SpreadCode_PA</td>
<td>OCNS spreading code array</td>
<td>{3, 4, 5, 6}</td>
<td>int array</td>
<td>[1, 16]</td>
<td></td>
</tr>
<tr>
<td>FilterLength</td>
<td>length of raised cosine filters in number of symbols</td>
<td>16</td>
<td>int</td>
<td></td>
<td>[1, ∞)</td>
</tr>
<tr>
<td>SystemDelay</td>
<td>total system delay in symbols including delay caused by filters</td>
<td>16</td>
<td>int</td>
<td></td>
<td>[1, ∞)</td>
</tr>
<tr>
<td>IgnoreNumber</td>
<td>ignored subframe numbers</td>
<td>4</td>
<td>int</td>
<td></td>
<td>[1, ∞)</td>
</tr>
<tr>
<td>PowerThreshold</td>
<td>power threshold for channel estimation</td>
<td>0</td>
<td>real</td>
<td></td>
<td>(0, ∞)</td>
</tr>
<tr>
<td>PhyChNum_SA</td>
<td>physical channel allocation configuration</td>
<td>0 1 0 0 0 0</td>
<td>int array</td>
<td>[0, 16] for Downlink, [0, 2] for Uplink</td>
<td></td>
</tr>
</tbody>
</table>
1. This subnetwork is used to implement 12.2k UL JD receiver.

The schematic for this subnetwork is shown in Figure 7-3.

### 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>InRF</td>
<td>input signals</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>output</td>
<td>output</td>
<td>int</td>
</tr>
</tbody>
</table>

† The FCarrier parameter sets the internal oscillator frequency used for demodulation. Setting FCarrier to -1 will use the input signal characterization frequency as the internal oscillator frequency.
Receivers

2. Please refer to the TDSCDMA_12_2_DL_RakeReceiver for the description of the receiver structure and algorithm.

References


TDSCDMA_12_2_UL_RakeReceiver

Description  Uplink rake receiver
Library  TDSCDMA, Receiver
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>FCarrier</td>
<td>carrier frequency</td>
<td>1900MHz</td>
<td>Hz</td>
<td>real</td>
<td>[-1] or (0, ∞)†</td>
</tr>
<tr>
<td>AWGN</td>
<td>AWGN channel or not: No, Yes</td>
<td></td>
<td></td>
<td>enum</td>
<td></td>
</tr>
<tr>
<td>SamplesPerSymbol</td>
<td>samples per symbol</td>
<td>8</td>
<td>int</td>
<td>[1, 32]</td>
<td></td>
</tr>
<tr>
<td>SlotIndex</td>
<td>slot index</td>
<td>2</td>
<td>int</td>
<td>[1, 6]</td>
<td></td>
</tr>
<tr>
<td>BasicMidambleID</td>
<td>index of basic midamble</td>
<td>0</td>
<td>int</td>
<td>[0, 127]</td>
<td></td>
</tr>
<tr>
<td>K</td>
<td>maximum number of midamble shifts in a cell</td>
<td>16</td>
<td>int</td>
<td>(2, 4, 6, 8, 10, 12, 14, 16)</td>
<td></td>
</tr>
<tr>
<td>MidambleID</td>
<td>midamble index</td>
<td>1</td>
<td>int</td>
<td>[1, K]</td>
<td></td>
</tr>
<tr>
<td>SpreadCode</td>
<td>spreading code for the DPCH</td>
<td>1</td>
<td>int</td>
<td>[1, 16]</td>
<td></td>
</tr>
<tr>
<td>FilterLength</td>
<td>length of raised cosine filters in number of symbols</td>
<td>16</td>
<td>int</td>
<td>[1, ∞)</td>
<td></td>
</tr>
<tr>
<td>SystemDelay</td>
<td>total system delay in symbols including delay caused by filters</td>
<td>16</td>
<td>int</td>
<td>[1, ∞)</td>
<td></td>
</tr>
<tr>
<td>IgnoreNumber</td>
<td>ignored subframe numbers</td>
<td>4</td>
<td>int</td>
<td>[1, ∞)</td>
<td></td>
</tr>
<tr>
<td>PowerThreshold</td>
<td>power threshold for channel estimation</td>
<td>0</td>
<td>real</td>
<td>(0, ∞)</td>
<td></td>
</tr>
<tr>
<td>PhyChNum_SA</td>
<td>physical channel allocation configuration</td>
<td>0 0 1 0 0 0 0</td>
<td>int array</td>
<td>[0, 16] for Downlink, [0, 2] for Uplink</td>
<td></td>
</tr>
<tr>
<td>MaxPhyChNum</td>
<td>sum of allocated physical channel in all slots</td>
<td>1</td>
<td>int</td>
<td>[1, 112]</td>
<td></td>
</tr>
</tbody>
</table>
Receivers

<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>ModType_SA</td>
<td>type of modulation of all slots, 0 for QPSK, 1 for 8PSK</td>
<td>0 0 0 0 0 0</td>
<td>int array</td>
<td>(0, 1)</td>
<td></td>
</tr>
<tr>
<td>TFCLSA</td>
<td>allocated TFCL transmitted active slots configuration</td>
<td>0 0 1 0 0 0 0</td>
<td>int array</td>
<td>(0, 1)</td>
<td></td>
</tr>
<tr>
<td>TFCLLength_SA</td>
<td>length of TFCL of all slots</td>
<td>0 0 1 6 0 0 0 0</td>
<td>int array</td>
<td>(0, 4,8,16,32) for QPSK, (0,6,12,24,48) for 8PSK</td>
<td></td>
</tr>
<tr>
<td>SS_TPC_SA</td>
<td>type of SS and TPC of all slots</td>
<td>2 2 1 2 2 2 2</td>
<td>int array</td>
<td>(1, 2,3)</td>
<td></td>
</tr>
<tr>
<td>MinSF_PA</td>
<td>minimum spreading factor array corresponding to allocated physical channels</td>
<td>8</td>
<td>int array</td>
<td>(1,16) for Downlink, (1,2,4,8,16) for Uplink</td>
<td></td>
</tr>
</tbody>
</table>

† The FCarrier parameter sets the internal oscillator frequency used for demodulation. Setting FCarrier to -1 will use the input signal characterization frequency as the internal oscillator frequency.

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>InRF</td>
<td>input signals</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>output</td>
<td>output</td>
<td>int</td>
</tr>
</tbody>
</table>

Notes/Equations

1. This subnetwork is used to implement 12.2k UL Rake receiver.

The schematic for this subnetwork is shown in Figure 7-4.
2. Please refer to the TDSCDMA_12_2_DL_RakeReceiver for the description of the receiver structure and algorithm.

References


Receivers

**TDSCDMA_A_Generator**

---

**Description**
Matrix A generator

**Library**
TDSCDMA, Receiver

**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>MidambleID</td>
<td>midamble index</td>
<td>1</td>
<td>int</td>
<td>([1, K])</td>
</tr>
<tr>
<td>ScrambleCode</td>
<td>index of scramble code</td>
<td>0</td>
<td>int</td>
<td>([0, 127])</td>
</tr>
<tr>
<td>SpreadFactor</td>
<td>spreading factor</td>
<td>16</td>
<td>int</td>
<td>([1, 2, 4, 8, 16])</td>
</tr>
<tr>
<td>SpreadCode</td>
<td>index of OVSF code</td>
<td>1</td>
<td>int</td>
<td>([1, \text{SpreadFactor}])</td>
</tr>
<tr>
<td>K</td>
<td>maximum number of midamble shifts in a cell</td>
<td>16</td>
<td>int</td>
<td>([2, 4, 6, 8, 10, 12, 14, 16])</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>h</td>
<td>channel impulse response</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>A</td>
<td>matrix A</td>
<td>complex matrix</td>
</tr>
</tbody>
</table>

---

**Notes/Equations**

1. This subnetwork is used to generate matrix A, which is used in the Rake or JD receiver.

   The schematic for this subnetwork is shown in Figure 7-5.
2. Matrix A is illustrated in Figure 7-6.

W is the estimation window length, M denote the spreading factor, N denote the number of symbols per slot, b is the vector generated by convoluting the channel impulse response with the spreading code.

References

Receivers

TDSCDMA_b_k_Generator

Description  Vector b_k generator
Library  TDSCDMA, Receiver
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>SpreadFactor</td>
<td>spreading factor</td>
<td>16</td>
<td>int</td>
<td>(1, 2, 4, 8, 16)</td>
</tr>
<tr>
<td>K</td>
<td>maximum number of midamble shifts in a cell</td>
<td>16</td>
<td>int</td>
<td>(2, 4, 6, 8, 10, 12, 14, 16)</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>h_k</td>
<td>channel impulse response</td>
<td>complex</td>
</tr>
<tr>
<td>2</td>
<td>OVSF</td>
<td>OVSF code</td>
<td>complex</td>
</tr>
<tr>
<td>3</td>
<td>scrb</td>
<td>scramble code</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>b_k</td>
<td>vector b_k</td>
<td>complex</td>
</tr>
</tbody>
</table>

Notes/Equations

1. This subnetwork is used to generate vector b_k, which is used to generate matrix A.
   The schematic for this subnetwork is shown in Figure 7-7.
2. Let W = estimation window length, M = spreading factor and N = number of symbols per slot.
The output b is the convolution of c with h, where
\[ c = (c_1 c_2 c_3 \ldots c_M)^T \]
is the spreading code,
\[ h = (h_1 h_2 h_3 \ldots h_W)^T \]
is the channel impulse response.
Each firing, M spreading code tokens, M scramble code tokens, and W channel impulse response tokens are consumed; M+W-1 output tokens are produced.

References
TDSCDMA_ChannelEstimation

**Description**  Channel estimation

**Library**  TDSCDMA, Receiver

**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>BasicMidambleID</td>
<td>index of basic midamble</td>
<td>0</td>
<td>int</td>
<td>[0, 127]</td>
</tr>
<tr>
<td>AWGN</td>
<td>AWGN channel or not: No, Yes</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PowerThreshold</td>
<td>power threshold for channel estimation</td>
<td>0</td>
<td>real</td>
<td>(0, ∞)</td>
</tr>
<tr>
<td>K</td>
<td>maximum number of midamble shifts in a cell</td>
<td>16</td>
<td>int</td>
<td>{2, 4, 6, 8, 10, 12, 14, 16}</td>
</tr>
<tr>
<td>IgnoreNumber</td>
<td>number of slots to be ignored</td>
<td>4</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>Mid</td>
<td>received midamble</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>h</td>
<td>channel impulse response</td>
<td>complex</td>
</tr>
</tbody>
</table>

**Notes/Equations**

1. This subnetwork is used to estimate channel impulse response.

   The schematic for this subnetwork is shown in Figure 7-8.
Each firing, 144 tokens are consumed, while 128 tokens are produced.

The channel impulse response is calculated by applying FFT. The advantage of FFT/IFFT is the circular characteristic of midamble. After 3 FFT/IFFT the channel impulse response for all the users can be determined. The algorithm is described as follows:

Figure 7-8. TDSCDMA_ChannelEstimation Schematic
Receivers

\[ h(1:128) = 128\text{FFT}(128\text{FFT}(\text{midamble\_data}(16:144)))/128\text{FFT}(\text{basic\_midamble\_code}(16:144))) \]

2. If channel type is AWGN, there is only one path, so only the path with maximum magnitude is selected. If channel type is not AWGN, PowerThreshold is used to refined the estimation; then

\[
h_{\text{refined}}(i) = \begin{cases} 
0; & \text{if}\left( h_{\text{Max}}^2 - h^2(i) \right) \geq \text{Power\_Threshold} \\
\text{h}(i); & \text{else}
\end{cases}
\]

References

TDSCDMA_JointDetection

Description Joint detection receiver
Library TDSCDMA, Receiver
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>K</td>
<td>maximum number of midamble shifts in a cell</td>
<td>16</td>
<td>int</td>
<td>(2, 4, 6, 8, 10, 12, 14, 16)</td>
</tr>
<tr>
<td>P</td>
<td>numbers of equivalent channels with spreading factor 16</td>
<td>2</td>
<td>int</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>received signal</td>
<td>complex</td>
</tr>
<tr>
<td>2</td>
<td>A</td>
<td>Matrix A</td>
<td>complex matrix</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</td>
<td>complex</td>
</tr>
</tbody>
</table>

Notes/Equations

1. This subnetwork is used to implement core algorithm of joint detection receiver. The schematic for this subnetwork is shown in Figure 7-9.
2. Let $e = A \cdot d + n$, where $e = (e_1, e_2, \ldots, e_{N \cdot M + W - 1})^T$ is the received sequence, $n = (n_1, n_2, \ldots, n_{N \cdot M + W - 1})^T$ is the noise sequence, $d = (d_1, d_2, \ldots, d_N)^T$ is the symbol sequence, $A$ is the transfer matrix defined in [1] and TDSCDMA_A_Generator, $W$ is the estimation window length, $M$ denotes the spreading factor and $N$ denotes the number of symbols per slot. Then the zero forcing joint detection receiver could be given by $d^*_D = (A^H A)^{-1} A^H e$. The matched filters maximize the output SNR, while the zero forcing joint detection eliminates the multi-user interference to obtain unbiased estimates. Interference results in SNR degradation.

References

TDSCDMA_Rake

Description  Rake receiver
Library   TDSCDMA, Receiver
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>K</td>
<td>maximum number of midamble shifts in a cell</td>
<td>16</td>
<td>int</td>
<td>(2, 4, 6, 8, 10, 12, 14, 16)</td>
</tr>
<tr>
<td>P</td>
<td>numbers of equivalent channels with spreading factor 16</td>
<td>2</td>
<td>int</td>
<td>[1, 16]</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 signal</td>
<td>complex</td>
</tr>
<tr>
<td>2</td>
<td>A</td>
<td>Matrix A</td>
<td>complex matrix</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</td>
<td>complex</td>
</tr>
</tbody>
</table>

Notes/Equations

1. This subnetwork is used to implement core algorithm of Rake receiver.

   The schematic for this subnetwork is shown in Figure 7-10.
2. Let \( e = A \cdot d + n \), where \( e = (e_1, e_2, e_3, \ldots, e_{N \cdot M + W - 1})^T \) is the received sequence, 
\( n = (n_1, n_2, n_3, \ldots, n_{N \cdot M + W - 1})^T \) is the noise sequence, \( d = (d_1, d_2, d_3, \ldots, d_N)^T \) is the symbol sequence, \( A \) is the transfer matrix defined in [1] and TDSCDMA_A_Generator, \( W \) is the estimation window length, \( M \) denotes the spreading factor and \( N \) denotes the number of symbols per slot. Then the matched filter receiver or RAKE receiver could be given by 
\( d_{\text{Rake}} = A^H e \).

References

Chapter 8: Signal Sources
**Signal Sources**

**TDSCDMA_DL_RF**

**Description**  
TDSCDMA downlink RF signal source

**Library**  
TDSCDMA, Signal Sources

**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>ROut</td>
<td>output resistance</td>
<td>DefaultROut</td>
<td>Ohm</td>
<td>real</td>
<td>(0, ∞)</td>
</tr>
<tr>
<td>FCarrier</td>
<td>carrier frequency</td>
<td>1900MHz</td>
<td>Hz</td>
<td>real</td>
<td>(0, ∞)</td>
</tr>
<tr>
<td>Power</td>
<td>modulator output power</td>
<td>10W</td>
<td>W</td>
<td>real</td>
<td>(0, ∞)</td>
</tr>
<tr>
<td>SlotIndex</td>
<td>index of slot</td>
<td>6</td>
<td>int</td>
<td></td>
<td>[0, 2,3,4,5,6]</td>
</tr>
<tr>
<td>MidambleAllocScheme</td>
<td>midamble allocation scheme: UE_Specific, Common, Default</td>
<td>Default</td>
<td>enum</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BasicMidambleID</td>
<td>index of basic midamble</td>
<td>0</td>
<td>int</td>
<td></td>
<td>[0, 127]</td>
</tr>
<tr>
<td>K</td>
<td>maximum number of midamble shifts in a cell</td>
<td>16</td>
<td>int</td>
<td></td>
<td>[2, 4,6,8,10,12,14,16]</td>
</tr>
<tr>
<td>MidambleID1</td>
<td>index of midamble for the first DPCH</td>
<td>1</td>
<td>int</td>
<td></td>
<td>[1, K]</td>
</tr>
<tr>
<td>MidambleID2</td>
<td>index of midamble for the second DPCH</td>
<td>2</td>
<td>int</td>
<td></td>
<td>[1, K]</td>
</tr>
<tr>
<td>SpreadCode1</td>
<td>index of spread code for the first DPCH</td>
<td>1</td>
<td>int</td>
<td></td>
<td>[1, 16]</td>
</tr>
<tr>
<td>SpreadCode2</td>
<td>index of spread code for the second DPCH</td>
<td>2</td>
<td>int</td>
<td></td>
<td>[1, 16]</td>
</tr>
<tr>
<td>PhyCHNum</td>
<td>number of channelization codes used in a timeslot</td>
<td>2</td>
<td>int</td>
<td></td>
<td>[1, 16]</td>
</tr>
<tr>
<td>SyncCode</td>
<td>index of basic synchronization code</td>
<td>0</td>
<td>int</td>
<td></td>
<td>[0, 31]</td>
</tr>
<tr>
<td>ModPhase</td>
<td>type of modulation quadruples: S1, S2</td>
<td>S1</td>
<td>enum</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DwPCH_Gain</td>
<td>Gain for DwPCH</td>
<td>1</td>
<td>int</td>
<td></td>
<td>(0, ∞)</td>
</tr>
</tbody>
</table>
1. This TD-SCDMA signal source generates a 12.2 kbps downlink (DL) RF signal with two dedicated physical channels (DPCH) and one downlink pilot channel (DwPCH).

To use this source, one typically needs to only set the RF carrier frequency (FCarrier) and power (Power).

Specific TD-SCDMA signal characteristics may be set, as may be required by a project system engineer, by setting parameters FilterLength, ModPhase, MidambleAllocScheme, SlotIndex, BasicMidambleD, MidambleD1, MidambleD2, K, SpreadCode1, SpreadCode2, DwPCH_Gain and SyncCode.

2. This signal source is composed of a DSP section and RF modulo as shown in the Figure 8-1.

The RF output from the signal source is at the frequency specified (FCarrier), with the specified source resistance (ROut) and power (Power).

---

### 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>OutRF</td>
<td>output signals</td>
<td>timed</td>
</tr>
<tr>
<td>2</td>
<td>TFCI</td>
<td>transport format combination indicator bits</td>
<td>int</td>
</tr>
<tr>
<td>3</td>
<td>SS</td>
<td>information bits for uplink synchronization control</td>
<td>int</td>
</tr>
<tr>
<td>4</td>
<td>TPC</td>
<td>transmit power control bits</td>
<td>int</td>
</tr>
<tr>
<td>5</td>
<td>Data</td>
<td>information data bits</td>
<td>multiple int</td>
</tr>
</tbody>
</table>

### Notes/Equations

1. This TD-SCDMA signal source generates a 12.2 kbps downlink (DL) RF signal with two dedicated physical channels (DPCH) and one downlink pilot channel (DwPCH).

To use this source, one typically needs to only set the RF carrier frequency (FCarrier) and power (Power).

Specific TD-SCDMA signal characteristics may be set, as may be required by a project system engineer, by setting parameters FilterLength, ModPhase, MidambleAllocScheme, SlotIndex, BasicMidambleD, MidambleD1, MidambleD2, K, SpreadCode1, SpreadCode2, DwPCH_Gain and SyncCode.

2. This signal source is composed of a DSP section and RF modulo as shown in the Figure 8-1.

The RF output from the signal source is at the frequency specified (FCarrier), with the specified source resistance (ROut) and power (Power).
3. This TD-SCDMA downlink signal source model is compatible with Agilent Signal Studio software option 411 for transmitter test. For your reference, the website of Signal Studio for TD-SCDMA is at http://we.home.agilent.com/cgi-bin/bvpub/agilent/Product/cp_Product.jsp?LANGUAGE_CODE=eng&COUNTRY_CODE=ZZ&NAV_ID=-12046.0.00&PID=find/signalstudio

4. Regarding the TD-SCDMA signal frame structure, one frame consists of two subframes. The subframe structure is illustrated in Figure 8-2. As can be seen, each subframe consists of 7 time slots (TS), and one downlink pilot time slot (DwPTS), one guard period (GP) and one uplink pilot time slot (UpPTS). Each time slot can transmit DPCH signals.

A TD-SCDMA chip has a sampling rate of 1.28 MHz.
For example, two DPCH signals in DPCH1 and DPCH2 are transmitted in TS0 shown in Figure 8-2.

Figure 8-2. SubFrame Structure of 12.2 kbps DL Channel

5. In general, changing the TD-SCDMA downlink source parameters settings from their default value will affect various transmitter measurements, including power, envelope, CCDF, and spectrum.

6. Parameter details.

- **SamplesPerSymbol** sets the number of samples in a chip. The default value of this parameter is set to 8 to display results properly under settings based on 3GPP NTDD standard.

  Set this value to a larger value if a simulation frequency bandwidth for this signal wider than $8 \times 1.28$ MHz is desired.

  Set this value to a smaller value for faster simulation times, but with lower signal fidelity.

- **FilterLength** shows root raised-cosine (RRC) filter length in chips. The default value of this parameter is set to 12 to transmit a TD-SCDMA downlink signal properly in both time and frequency domains based on 3GPP NTDD standard [1-3].
Signal Sources

Set this value to a smaller value for faster simulation times, but at the cost of lower signal fidelity.

- ModPhase is used to select the phase quadruples of DwPTS for different phase rotation pattern. In Signal Studio, a Rotation Phase parameter is used to select the phase quadruples.

There are two different phase quadruples, S1 and S2 specified by 3GPP NTDD standard [3]. A quadruple always starts with an even signal frame number. Table 8-1 describes the quadruples, where P-CCPCH is the primary common control physical channel.

| Table 8-1. Phase Modulation Sequences for Downlink Synchronization Code |
|----------------|-----------------|-----------------|
| Name | Phase Quadruple | Description |
| S1 | 135,45,225,135 | There is a P-CCPCH in the next 4 sub-frames |
| S2 | 315,225,315,45 | There is no P-CCPCH in the next 4 sub-frames |

- MidambleAllocScheme is used to select the midamble allocation scheme. There are three midamble allocation schemes based on 3GPP NTDD standard [1,2].

  **UE_Specific**: a UE-specific midamble allocation for downlink and uplink is explicitly assigned by higher layers.

  **Default**: the midamble allocation for downlink and uplink is assigned by layer 1 depending on associated channelization code.

  **Common**: the downlink midamble allocation is assigned by layer 1 depending on the number of channelization codes currently present in the downlink time slot.

To set MidambleAllocScheme parameter based on 3GPP NTDD standard [1], related parameters must also be set:

- if MidambleAllocScheme = UE_Specific, the BasicMidambleID, K and MidambleID parameters are used to specify which midamble is exported.

- if MidambleAllocScheme = Common, only the BasicMidambleID, K are used to specify which midamble is exported, the MidambleID parameter is ignored.
• if MidambleAllocScheme = Default, only the BasicMidamble D, K are used to specify which midamble is exported, the Midamble D parameter is ignored.

• SlotIndex parameter is used to select which slot signal in the subframe will be transmitted.

• BasicMidamble D sets the basic midamble code ID. The basic midamble code is used for training sequences for uplink and downlink channel estimation, power measurements and maintaining uplink synchronization. There are 128 different sequences; BasicMidamble D can be set from 0 to 127. In Signal Studio, Basic Midamble ID code has the same meaning as this parameter.

• K is the maximum number of different midamble shifts in a cell that can be decided by maximum users in the cell for current time slot.

• Midamble D1 and Midamble D2 set indices of midambles for the first and second DPCH, respectively. Midambles of different users active in the same cell and the same time slot are cyclically shifted versions of one basic midamble code.

Let P = 128, the length of basic midamble, then \( W = \left\lfloor \frac{P}{K} \right\rfloor \), is the shift between midambles and \( \lfloor x \rfloor \) denotes the largest number less or equal to x. Midamble D range is from 1 to K.

Midamble D and K together correspond to parameter of Midamble Offset in Signal Studio for Timeslot setup. Midamble Offset = Midamble D \times W.

• SpreadCode1 and SpreadCode2 set spread code indices for the first and second DPCH, respectively. For this signal source, the spreading factor is 16. In Signal Studio, Channelization code for Time slot setup has the same meaning as SpreadCode1 and SpreadCode2.

• DwPCH_Gain sets the gain of DwPCH relative to DPCH. In Signal Studio, there are dialog boxes with dB unit for each DwPCH to set the gain of DwPCH relative to DPCH.

• SyncCode sets the downlink pilot synchronization sequence (SYNC-DL). Downlink pilot synchronization is used for downlink synchronization and cell initial search. There are 32 different SYNC-DL code groups that are used to distinguish base stations.

DwPTS is composed of 64 chips of a complex SYNC_DL sequence:

\[ S = (s_1, s_2, \ldots, s_{64}) \]
Signal Sources

and 32 chips of guard period.

To generate the complex SYNC_DL code, the basic SYNC_DL code

\[ s = (s_1, s_2, ..., s_{64}) \]

is used.

There are 32 different basic SYNC_DL codes for the entire system. The relation between \( s \) and \( s_i \) is given by:

\[ s_i = (j)^{v_i} s_i \text{ where } v_i \in \{1, -1\}, i = 1, ..., 64 \]

Therefore, the elements \( s_i \) of \( s \) are alternating real and imaginary.

In Signal Studio, SYNC Code is used to set the downlink pilot code.

References


TDSCDMA_DL_Src

Description  TDSCDMA downlink signal source
Library  TDSCDMA, Signal Sources
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>SlotIndex</td>
<td>index of slot</td>
<td>6</td>
<td>int</td>
<td>(0, 2, 3, 4, 5, 6)</td>
</tr>
<tr>
<td>MidambleAllocScheme</td>
<td>midamble allocation scheme: UE_Specific, Common, Default</td>
<td>Default</td>
<td>enum</td>
<td></td>
</tr>
<tr>
<td>BasicMidambleID</td>
<td>index of basic midamble</td>
<td>0</td>
<td>int</td>
<td>[0, 127]</td>
</tr>
<tr>
<td>K</td>
<td>maximum number of midamble shifts in a cell</td>
<td>16</td>
<td>int</td>
<td>(2, 4, 6, 8, 10, 12, 14, 16)</td>
</tr>
<tr>
<td>MidambleID1</td>
<td>index of midamble for the first DPCH</td>
<td>1</td>
<td>int</td>
<td>[1, K]</td>
</tr>
<tr>
<td>MidambleID2</td>
<td>index of midamble for the second DPCH</td>
<td>2</td>
<td>int</td>
<td>[1, K]</td>
</tr>
<tr>
<td>SpreadCode1</td>
<td>index of spread code for the first DPCH</td>
<td>1</td>
<td>int</td>
<td>[1, 16]</td>
</tr>
<tr>
<td>SpreadCode2</td>
<td>index of spread code for the second DPCH</td>
<td>2</td>
<td>int</td>
<td>[1, 16]</td>
</tr>
<tr>
<td>PhyChNum</td>
<td>number of channelization codes used in a timeslot</td>
<td>2</td>
<td>int</td>
<td>[1, 16]</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>TFCI</td>
<td>transport format combination indicator bits</td>
<td>int</td>
</tr>
<tr>
<td>2</td>
<td>SS</td>
<td>information bits for uplink synchronization control</td>
<td>int</td>
</tr>
<tr>
<td>3</td>
<td>TPC</td>
<td>transmit power control bits</td>
<td>int</td>
</tr>
<tr>
<td>4</td>
<td>input</td>
<td>input data</td>
<td>multiple int</td>
</tr>
</tbody>
</table>
Signal Sources

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>output data</td>
<td>complex</td>
</tr>
</tbody>
</table>

Notes/Equations

1. This subnetwork generates a downlink signal source with 12.2 kbps that includes two DPCHs.
   The schematic for this subnetwork is shown in Figure 8-3.

   ![Figure 8-3. TDSCDMA_DL_Src Schematic](image)

2. The frame structure is illustrated in Figure 8-4.

   ![Figure 8-4. Frame Structure of 12.2 kbps Downlink Channel](image)

References

TDSCDMA_OCNS

Description  Flexible OCNS generator
Library  TDSCDMA, Signal Sources
Class  SDFTDSCDMA_OCNS
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>ModType_PA</td>
<td>type of modulation corresponding to allocated physical channels</td>
<td>0</td>
<td>int array</td>
<td>[0, 1]</td>
</tr>
<tr>
<td>SpreadFactor_PA</td>
<td>spreading factor corresponding to allocated physical channels</td>
<td>1</td>
<td>int array</td>
<td>{1, 2, 4, 8, 16}</td>
</tr>
<tr>
<td>SpreadCode_PA</td>
<td>index of OVSF code corresponding to allocated physical channels</td>
<td>1</td>
<td>int array</td>
<td>{1, SpreadFactor}</td>
</tr>
<tr>
<td>BasicMidambleID</td>
<td>index of basic midamble</td>
<td>1</td>
<td>int</td>
<td>[0, 127]</td>
</tr>
<tr>
<td>K</td>
<td>maximum number of midamble shifts in a cell</td>
<td>16</td>
<td>int</td>
<td>{2, 4, 6, 8, 10, 12, 14, 16}</td>
</tr>
<tr>
<td>MidambleID_PA</td>
<td>index of midamble corresponding to allocated physical channels</td>
<td>1</td>
<td>int array</td>
<td>{1, K}</td>
</tr>
<tr>
<td>Gain_PA</td>
<td>gain setting array corresponding to allocated physical channels</td>
<td>1.0</td>
<td>real array</td>
<td>(0, ∞)</td>
</tr>
<tr>
<td>SlotIndex</td>
<td>allocated active slots configuration</td>
<td>1</td>
<td>int</td>
<td>[0, 6]</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>DataOut</td>
<td>output data</td>
<td>multiple complex</td>
</tr>
</tbody>
</table>
Signal Sources

Notes/Equations

1. This model is a flexible orthogonal channel noise simulator. The number of DPCH is determined by the dimension of ModType_SA. However, the dimensions of all array parameter must be the same.

   Each firing, 6400 DataOut tokens are produced.

2. SlotIndex indicates the index of slot in which the physical channel will be transmitted.

3. ModType_PA is the modulation mapping scheme of the data bits, 0 for QPSK, 1 for 8PSK.

4. The elements of SpreadFactor_PA and SpreadCode_PA are the spreading factor and index of spreading code for each physical channel.

5. The index of scramble code is the same as BasicMidambleID.

6. The midamble of each physical channel is determined by UE_Specific based on K and MidambleID_PA settings.

7. Gain_PA determines the gain of each physical channel.

References


TDSCDMA_SlotSrc

**Description** Flexible SubFrame generator

**Library** TDSCDMA, Signal Sources

**Class** SDFTDSCDMA_SlotSrc

**Derived From** TDSCDMA_CCTrCH_MuxBase

**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>Link</td>
<td>link selection: Downlink, Uplink</td>
<td>Uplink</td>
<td>enum</td>
<td></td>
</tr>
<tr>
<td>SlotIndex</td>
<td>allocated active slots configuration</td>
<td>1</td>
<td>int</td>
<td>[0, 6]</td>
</tr>
<tr>
<td>ChannelState_PA</td>
<td>active status for each slots</td>
<td>1</td>
<td>int array</td>
<td>(0, 1)</td>
</tr>
<tr>
<td>ModType_PA</td>
<td>type of modulation corresponding to allocated physical channels</td>
<td>0</td>
<td>int array</td>
<td>(0, 1)</td>
</tr>
<tr>
<td>SpreadFactor_PA</td>
<td>spreading factor corresponding to allocated physical channels</td>
<td>16</td>
<td>int array</td>
<td>(1, 2,4,8,16)</td>
</tr>
<tr>
<td>SpreadCode_PA</td>
<td>index of OVSF code corresponding to allocated physical channels</td>
<td>16</td>
<td>int array</td>
<td>[1, SpreadFactor]</td>
</tr>
<tr>
<td>BasicMidambleID</td>
<td>index of basic midamble</td>
<td>1</td>
<td>int</td>
<td>[0, 127]</td>
</tr>
<tr>
<td>K</td>
<td>maximum number of midamble shifts in a cell</td>
<td>16</td>
<td>int</td>
<td>(2, 4,6,8,10,12,14,16)</td>
</tr>
<tr>
<td>UserID_PA</td>
<td>user id for each DPCH in a slot</td>
<td>1</td>
<td>int array</td>
<td>[1, K]</td>
</tr>
<tr>
<td>Gain_PA</td>
<td>gain setting array corresponding to allocated physical channels</td>
<td>1.0</td>
<td>real array</td>
<td>(0, ∞)</td>
</tr>
</tbody>
</table>
1. This model is a flexible channel signal simulator that generates signals for several physical channels in one specified time slot. The number of slots is determined by the SlotIndex parameter. The physical channels are specified by 1s in the ChannelState_PA parameter array; the maximum number of physical channels in one timeslot is 16. Dimensions of all other array parameters are determined by ChannelState_PA.

2. The output of this model is a subframe with one time slot data filled. Each firing, 6400 DataOut tokens are produced. The data of each physical channel is randomly generated, modulated, spread and scrambled.

3. In the TD-SCDMA signal frame structure, one frame consists of two subframes. The subframe structure is illustrated in Figure 8-5; each subframe consists of 7 time slots (TS), and one downlink pilot time slot (DwPTS), one guard period (GP) and one uplink pilot time slot (UpPTS). Each time slot transmits physical channel signals. Each physical channel signal is composed of 704 chips for data, TFCI and TPC, 144 chips midamble and 16 chips guard period.

   Figure 8-5 illustrates an output subframe with n physical channels in TS0.
4. Link sets uplink/downlink for each slot; the link is limited to downlink in TS0.

5. SlotIndex indicates the slot index in which the physical channels will be transmitted.

6. ChannelState_PA is an integer array indicating the on/off status of each physical channel; when the element of the parameter is set to 1, the corresponding physical channel is active, otherwise 0 for inactive.

7. ModType_PA determines the modulation mapping scheme of the data bits: 0 for QPSK and 1 for 8PSK.

8. After modulation, data is spread with corresponding spreading codes. The spreading factor of each physical channel is determined by SpreadFactor_PA, while the spreading code index is set by SpreadCode_PA.

9. BasicMidambleID sets the basic midamble code ID. There are 128 different sequences. Hence, BasicMidambleID can be set from 0 to 127.

10. UserID_PA sets the indices of midambles for the each PCH. Midambles of different users active in the same cell and the same time slot are cyclically shifted versions of one basic midamble code.

Let

\[ P = 128 \], the length of basic midamble; set \( K \) as the max midamble shift,
then

\[ W = \left\lfloor \frac{p}{K} \right\rfloor \] is the shift between midambles

\( \lfloor x \rfloor \) denotes the largest number less or equal to \( x \).

11. The midamble allocation schemes are based on 3GPP NTDD standard [1,2].

- **UE specific midamble allocation:** a UE specific midamble for downlink and uplink is explicitly assigned by higher layers
- **Default midamble allocation:** the midamble for downlink and uplink is assigned by layer 1 depending on associated channelization code.
- **Common midamble allocation:** the midamble for downlink is allocated by layer 1 depending on the number of channelization codes currently present in the downlink time slot.

In this model, the **UE Specific midamble allocation scheme** is used to generate midamble chips based on \( K \) and UserID_PA settings.

12. The index of scramble code is the same as BasicMidambleID.

13. **Gain_PA** determines the gain of each physical channel.

**References**


TDSCDMA_UL_RF

Description  TDSCDMA uplink RF signal source
Library  TDSCDMA, Signal Sources
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>ROut</td>
<td>output resistance</td>
<td>DefaultROut</td>
<td>Ohm</td>
<td>real</td>
<td>(0, ∞)</td>
</tr>
<tr>
<td>FCarrier</td>
<td>carrier frequency</td>
<td>1900MHz</td>
<td>Hz</td>
<td>real</td>
<td>(0, ∞)</td>
</tr>
<tr>
<td>Power</td>
<td>modulator output power</td>
<td>0.1W</td>
<td>W</td>
<td>real</td>
<td>(0, ∞)</td>
</tr>
<tr>
<td>SlotIndex</td>
<td>index of slot</td>
<td>2</td>
<td>int</td>
<td>[1, 6]</td>
<td></td>
</tr>
<tr>
<td>MidambleAllocScheme</td>
<td>midamble allocation scheme: UE_Specific, Default</td>
<td>Default</td>
<td>enum</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BasicMidambleID</td>
<td>index of basic midamble</td>
<td>0</td>
<td>int</td>
<td>[0, 127]</td>
<td></td>
</tr>
<tr>
<td>K</td>
<td>maximum number of midamble shifts in a cell</td>
<td>16</td>
<td>int</td>
<td>[2, 4, 6, 8, 10, 12, 14, 16]</td>
<td></td>
</tr>
<tr>
<td>MidambleID</td>
<td>index of midamble</td>
<td>1</td>
<td>int</td>
<td>[1, K]</td>
<td></td>
</tr>
<tr>
<td>SpreadCode</td>
<td>index of spread code</td>
<td>1</td>
<td>int</td>
<td>[1, 8]</td>
<td></td>
</tr>
<tr>
<td>SamplesPerSymbol</td>
<td>samples per symbol period</td>
<td>8</td>
<td>int</td>
<td>[1, 32]</td>
<td></td>
</tr>
<tr>
<td>FilterLength</td>
<td>length of raised cosine filters in number of symbols</td>
<td>16</td>
<td>int</td>
<td>(0, ∞)</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>OutRF</td>
<td>output signals</td>
<td>timed</td>
</tr>
<tr>
<td>2</td>
<td>TFCI</td>
<td>transport format combination indicator bits</td>
<td>int</td>
</tr>
<tr>
<td>3</td>
<td>SS</td>
<td>information bits for uplink synchronization control</td>
<td>int</td>
</tr>
</tbody>
</table>
Signal Sources

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>TPC</td>
<td>Transmit power control bits</td>
<td>int</td>
</tr>
<tr>
<td>5</td>
<td>Data</td>
<td>Input information data bits</td>
<td>int</td>
</tr>
</tbody>
</table>

Notes/Equations

1. This TD-SCDMA signal source generates a 12.2 kbps uplink (UL) RF signal with one dedicated physical channel (DPCH).

2. To use this source, one typically needs to only set the RF carrier frequency (FCarrier) and power (Power).

Specific TD-SCDMA signal characteristics may be set, as may be required by a project system engineer, by setting parameters FilterLength, MidambleAllocScheme, SlotIndex, BasicMidambleID, MidambleID, K and SpreadCode.

3. This signal source is composed of a DSP section, RF modulator and output source resistor as shown in Figure 8-6.

The RF output from the signal source is at the frequency specified (FCarrier), with the specified source resistance (ROut) and power (Power).
4. This TD-SCDMA uplink signal source model is compatible with Agilent Signal Studio software option 411 for transmitter test. For your reference, the website of Signal Studio for TD-SCDMA is at http://we.home.agilent.com/cgi-bin/bvpub/agilent/Product/cp_Product.jsp?LANGUAGE_CODE=eng&COUNTRY_CODE=ZZ&NAV_ID=12046.0.00&j PID=find/signalstudio

This partially coded TD-SCDMA signal source per 3GPP NTDD is almost identical to TD-SCDMA TSM defined for Signal Studio.

5. Regarding the TD-SCDMA signal frame structure, one frame consists of two subframes. The subframe structure is illustrated in Figure 8-7. As can be seen, each subframe consists of 7 time slots (TS), and one downlink pilot time slot (DwPTS), one guard period (GP) and one uplink pilot time slot (UpPTS). Each time slot can transmit DPCH signals.
6. In general, changing the TD-SCDMA uplink source parameters settings from their default value will affect various transmitter measurements including Power, Envelope, CCDF, and Spectrum.

7. Parameter details

- **SamplesPerSymbol** sets the number of samples in a chip. The default value of this parameter is set to 8 to display results properly under settings based on 3GPP NTDD standard. Set this value to a larger value if a simulation frequency bandwidth for this signal wider than 8*1.28 MHz is desired. Set this value to a smaller value for faster simulation times, but at the cost of lower signal fidelity.

- **FilterLength** shows root raised-cosine (RRC) filter length in chips. The default value of this parameter is set to 12 to transmit TD-SCDMA uplink signals in both time and frequency domains according to 3GPP NTDD standard [1-3]. Set this value to a smaller value for faster simulation times, but at the cost of lower signal fidelity.

- **MidambleAllocScheme** is used to select the midamble allocation scheme. There are three midamble allocation schemes based on 3GPP NTDD standard [1,2].
**UE_Specific**: a UE-specific midamble allocation for downlink and uplink is explicitly assigned by higher layers.

**Default**: the midamble allocation for downlink and uplink is assigned by layer 1 depending on associated channelization code.

**Common**: the downlink midamble allocation is assigned by layer 1 depending on the number of channelization codes currently present in the downlink time slot.

To set MidambleAllocScheme parameter based on 3GPP NTDD standard [1], related parameters must also be set:

- if MidambleAllocScheme = UE_Specific, the BasicMidambleID, K and MidambleID parameters are used to specify which midamble is exported.
- if MidambleAllocScheme = Common, only the BasicMidambleID, K are used to specify which midamble is exported, the MidambleID parameter is ignored.
- if MidambleAllocScheme = Default, only the BasicMidambleID, K are used to specify which midamble is exported, the MidambleID parameter is ignored.

- SlotIndex parameter is used to select which slot signal in the subframe will be transmitted.
- BasicMidambleID sets the basic midamble code ID. The basic midamble code is used for training sequences for uplink and downlink channel estimation, power measurements and maintaining uplink synchronization. There are 128 different sequences; BasicMidambleID can be set from 0 to 127. In Signal Studio, Basic Midamble ID code has the same meaning as this parameter.
- K is the maximum number of different midamble shifts in a cell that can be determined by maximum users in the cell for current time slot.
- MidambleID sets the index of midambles for DPCH. Midambles of different users active in the same cell and the same time slot are cyclically shifted versions of one basic midamble code.

Let \( P = 128 \), the length of basic midamble, then \( W = \left\lfloor \frac{P}{K} \right\rfloor \), is the shift between midambles and \( \lfloor x \rfloor \) denotes the largest number less or equal to \( x \). MidambleID range is from 1 to \( K \).

MidambleID and K together correspond to parameter of Midamble Offset in Signal Studio for Timeslot setup. Midamble Offset = MidambleID * W.
Signal Sources

- SpreadCode sets the spread code index for the DPCH. For this signal source, the spreading factor is 8. In Signal Studio, Channelization code for Time slot setup has the same meaning as SpreadCode.

References

TDSCDMA_UL_Src

Description  TDSCDMA uplink signal source  
Library  TDSCDMA, Signal Sources  
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>SlotIndex</td>
<td>index of slot</td>
<td>2</td>
<td>int</td>
<td>[1, 6]</td>
</tr>
<tr>
<td>MidambleAllocScheme</td>
<td>midamble allocation scheme: UE_Specific, Default</td>
<td></td>
<td>enum</td>
<td></td>
</tr>
<tr>
<td>BasicMidambleID</td>
<td>index of basic midamble</td>
<td>0</td>
<td>int</td>
<td>[0, 127]</td>
</tr>
<tr>
<td>K</td>
<td>maximum number of midamble shifts in a cell</td>
<td>16</td>
<td>int</td>
<td>{2, 4, 6, 8, 10, 12, 14, 16}</td>
</tr>
<tr>
<td>MidambleID</td>
<td>index of midamble</td>
<td>1</td>
<td>int</td>
<td>[1, K]</td>
</tr>
<tr>
<td>SpreadCode</td>
<td>index of spread code</td>
<td>1</td>
<td>int</td>
<td>[1, 8]</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>input data</td>
<td>int</td>
</tr>
<tr>
<td>2</td>
<td>TFCI</td>
<td>transport format combination indicator bits</td>
<td>int</td>
</tr>
<tr>
<td>3</td>
<td>SS</td>
<td>information bits for uplink synchronization control</td>
<td>int</td>
</tr>
<tr>
<td>4</td>
<td>TPC</td>
<td>transmit power control 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>5</td>
<td>Output</td>
<td>output data</td>
<td>complex</td>
</tr>
</tbody>
</table>

Notes/Equations
Signal Sources

1. This subnetwork generates an uplink signal source with 12.2 kbps that includes one DPCH.

   The schematic for this subnetwork is shown in Figure 8-8.

   ![Figure 8-8. TDSCDMA_UL_Src Schematic](image)

2. The frame structure is illustrated in Figure 8-9.

   ![Figure 8-9. Frame Structure of 12.2 kbps UL Channel](image)

References

## Index

**T**

<table>
<thead>
<tr>
<th>TDSCDMA_12_2_DL_JD_Receiver</th>
<th>7-2</th>
</tr>
</thead>
<tbody>
<tr>
<td>TDSCDMA_12_2_DL_RakeReceiver</td>
<td>7-6</td>
</tr>
<tr>
<td>TDSCDMA_12_2_UL_JD_Receiver</td>
<td>7-10</td>
</tr>
<tr>
<td>TDSCDMA_12_2_UL_RakeReceiver</td>
<td>7-13</td>
</tr>
<tr>
<td>TDSCDMA_1stDeIntlvr</td>
<td>5-2</td>
</tr>
<tr>
<td>TDSCDMA_1stIntlvr</td>
<td>5-4</td>
</tr>
<tr>
<td>TDSCDMA_2ndDeIntlvr</td>
<td>5-6</td>
</tr>
<tr>
<td>TDSCDMA_2ndIntlvr</td>
<td>5-8</td>
</tr>
<tr>
<td>TDSCDMA_A_Generator</td>
<td>7-16</td>
</tr>
<tr>
<td>TDSCMA_b_k_Generator</td>
<td>7-18</td>
</tr>
<tr>
<td>TDSCDMA_BER</td>
<td>3-2</td>
</tr>
<tr>
<td>TDSCDMA_BurstDeMux</td>
<td>4-2</td>
</tr>
<tr>
<td>TDSCDMA_BurstMux</td>
<td>4-4</td>
</tr>
<tr>
<td>TDSCDMA_ChannelEstimation</td>
<td>7-20</td>
</tr>
<tr>
<td>TDSCDMA_ChCoding</td>
<td>5-18</td>
</tr>
<tr>
<td>TDSCDMA_ChDecoding</td>
<td>5-21</td>
</tr>
<tr>
<td>TDSCDMA_CodeBkSeg</td>
<td>5-25</td>
</tr>
<tr>
<td>TDSCDMA_Constellation</td>
<td>3-3</td>
</tr>
<tr>
<td>TDSCDMA_CRC_Decoder</td>
<td>5-12</td>
</tr>
<tr>
<td>TDSCDMA_CRC_Encoder</td>
<td>5-15</td>
</tr>
<tr>
<td>TDSCDMA_DeCodeBkSeg</td>
<td>5-27</td>
</tr>
<tr>
<td>TDSCDMA_Demodulator</td>
<td>4-23</td>
</tr>
<tr>
<td>TDSCDMA_DePhyChMap</td>
<td>5-30</td>
</tr>
<tr>
<td>TDSCDMA_DePhyChSeg</td>
<td>5-32</td>
</tr>
<tr>
<td>TDSCDMA_DeRadioEqual</td>
<td>5-36</td>
</tr>
<tr>
<td>TDSCDMA_DeRadioSeg</td>
<td>5-38</td>
</tr>
<tr>
<td>TDSCDMA_DeRateMatch</td>
<td>5-41</td>
</tr>
<tr>
<td>TDSCDMA_DeSubFrameSeg</td>
<td>5-45</td>
</tr>
<tr>
<td>TDSCDMA_DL_RF</td>
<td>8-2</td>
</tr>
<tr>
<td>TDSCDMA_DL_Src</td>
<td>8-9</td>
</tr>
<tr>
<td>TDSCDMA_DPCH</td>
<td>6-2</td>
</tr>
<tr>
<td>TDSCDMA_DPCH_DataDeMux</td>
<td>4-6</td>
</tr>
<tr>
<td>TDSCDMA_DPCH_DataMux</td>
<td>4-14</td>
</tr>
<tr>
<td>TDSCDMA_DPCH_Mux</td>
<td>4-20</td>
</tr>
<tr>
<td>TDSCDMA_DwPCH</td>
<td>6-9</td>
</tr>
<tr>
<td>TDSCDMA_EVM</td>
<td>3-6</td>
</tr>
<tr>
<td>TDSCDMA_FPACH</td>
<td>6-12</td>
</tr>
<tr>
<td>TDSCDMA_FrameSync</td>
<td>3-13</td>
</tr>
<tr>
<td>TDSCDMA_FwdChannel</td>
<td>3-16</td>
</tr>
<tr>
<td>TDSCDMA_JointDetection</td>
<td>7-23</td>
</tr>
<tr>
<td>TDSCDMA_Midamble</td>
<td>4-25</td>
</tr>
<tr>
<td>TDSCDMA_Modulator</td>
<td>4-30</td>
</tr>
<tr>
<td>TDSCDMA_OCNS</td>
<td>8-11</td>
</tr>
<tr>
<td>TDSCDMA_OnePhyCh</td>
<td>4-33</td>
</tr>
<tr>
<td>TDSCDMA_OnePhyChDeMux</td>
<td>4-35</td>
</tr>
<tr>
<td>TDSCDMA_OVSF</td>
<td>4-37</td>
</tr>
<tr>
<td>TDSCDMA_PCCPCH</td>
<td>6-14</td>
</tr>
<tr>
<td>TDSCDMA_PhychMap</td>
<td>5-47</td>
</tr>
<tr>
<td>TDSCDMA_PhychSeg</td>
<td>5-49</td>
</tr>
<tr>
<td>TDSCDMA_PICH</td>
<td>6-16</td>
</tr>
<tr>
<td>TDSCDMA_PRACH</td>
<td>6-20</td>
</tr>
<tr>
<td>TDSCDMA_PsCH</td>
<td>6-22</td>
</tr>
<tr>
<td>TDSCDMA_PSCH_DataMux</td>
<td>4-40</td>
</tr>
<tr>
<td>TDSCDMA_RadioEqual</td>
<td>5-54</td>
</tr>
<tr>
<td>TDSCDMA_RadioSeg</td>
<td>5-56</td>
</tr>
<tr>
<td>TDSCDMA_Rake</td>
<td>7-25</td>
</tr>
<tr>
<td>TDSCDMA_RateMatch</td>
<td>5-58</td>
</tr>
<tr>
<td>TDSCDMA_RefCh</td>
<td>2-2</td>
</tr>
<tr>
<td>TDSCDMA_RefCh_RF</td>
<td>2-13</td>
</tr>
<tr>
<td>TDSCDMA_RefChDecoder</td>
<td>5-62</td>
</tr>
<tr>
<td>TDSCDMA_RevChannel</td>
<td>3-23</td>
</tr>
<tr>
<td>TDSCDMA_RF_CCDF</td>
<td>3-18</td>
</tr>
<tr>
<td>TDSCDMA_RF_PwrMeasure</td>
<td>3-20</td>
</tr>
<tr>
<td>TDSCDMA_RM_Cal</td>
<td>5-72</td>
</tr>
<tr>
<td>TDSCDMA_SCCPCH</td>
<td>6-25</td>
</tr>
<tr>
<td>TDSCDMA_Scramble</td>
<td>4-42</td>
</tr>
<tr>
<td>TDSCDMA_SlotSrc</td>
<td>8-13</td>
</tr>
<tr>
<td>TDSCDMA_SubFrameSeg</td>
<td>5-74</td>
</tr>
<tr>
<td>TDSCDMA_Sync</td>
<td>4-44</td>
</tr>
<tr>
<td>TDSCDMA_TFCI_Encoder</td>
<td>5-76</td>
</tr>
<tr>
<td>TDSCDMA_TrChDeMux</td>
<td>5-78</td>
</tr>
<tr>
<td>TDSCDMA_TrChMux</td>
<td>5-83</td>
</tr>
<tr>
<td>TDSCDMA_UL_RF</td>
<td>8-17</td>
</tr>
<tr>
<td>TDSCDMA_UL_Src</td>
<td>8-23</td>
</tr>
<tr>
<td>TDSCDMA_UpPCH</td>
<td>6-27</td>
</tr>
</tbody>
</table>

---

Index-1
Index-2