TMS320TCI648x DSP Turbo-Decoder Coprocessor 2 (TCP2)

User's Guide

Literature Number: SPRUE10A
May 2006–Revised June 2008

Contents

List of Figures

List of Tables

Read This First

About This Manual

Channel decoding of high bit-rate data channels found in third-generation (3G) cellular standards requires decoding of turbo-encoded data. The turbo-decoder coprocessor (TCP2) in some of the digital signal processor (DSPs) of the TMS320C6000™ DSP family has been designed to perform this operation for IS2000 and 3GPP wireless standards. This document describes the operation and programming of the TCP2.

Notational Conventions

This document uses the following conventions.

- Hexadecimal numbers are shown with the suffix h. For example, the following number is 40 hexadecimal (decimal 64): 40h.
- Registers in this document are shown in figures and described in tables.
  - Each register figure shows a rectangle divided into fields that represent the fields of the register. Each field is labeled with its bit name, its beginning and ending bit numbers above, and its read/write properties below. A legend explains the notation used for the properties.
  - Reserved bits in a register figure designate a bit that is used for future device expansion.
- The term "word" describes a 32-bit value. The term "halfword" describes a 16-bit value.

Related Documentation From Texas Instruments

The following documents describe the C6000™ devices and related support tools. Copies of these documents are available on the Internet at www.ti.com. *Tip:* Enter the literature number in the search box provided at www.ti.com.

SPRU189 — *TMS320C6000 DSP CPU and Instruction Set Reference Guide.* Describes the CPU architecture, pipeline, instruction set, and interrupts for the TMS320C6000 digital signal processors (DSPs).

SPRU198 — *TMS320C6000 Programmer's Guide.* Describes ways to optimize C and assembly code for the TMS320C6000™ DSPs and includes application program examples.

SPRU301 — *TMS320C6000 Code Composer Studio Tutorial.* Introduces the Code Composer Studio™ integrated development environment and software tools.

SPRU321 — *Code Composer Studio Application Programming Interface Reference Guide.* Describes the Code Composer Studio™ application programming interface (API), which allows you to program custom plug-ins for Code Composer.

SPRU871 — *TMS320C64x+ Megamodule Reference Guide.* Describes the TMS320C64x+ digital signal processor (DSP) megamodule. Included is a discussion on the internal direct memory access (IDMA) controller, the interrupt controller, the power-down controller, memory protection, bandwidth management, and the memory and cache.

Trademarks

TMS320C6000, C6000, Code Composer Studio are trademarks of Texas Instruments.

TMS320TCI648x DSP Turbo-Decoder Coprocessor 2

Channel decoding of high bit-rate data channels found in third-generation (3G) cellular standards requires decoding of turbo-encoded data. The turbo-decoder coprocessor (TCP2) in some of the digital signal processor (DSPs) of the TMS320C6000E DSP family has been designed to perform this operation for IS2000 and 3GPP wireless standards. This document describes the operation and programming of the TCP2.

1 Features

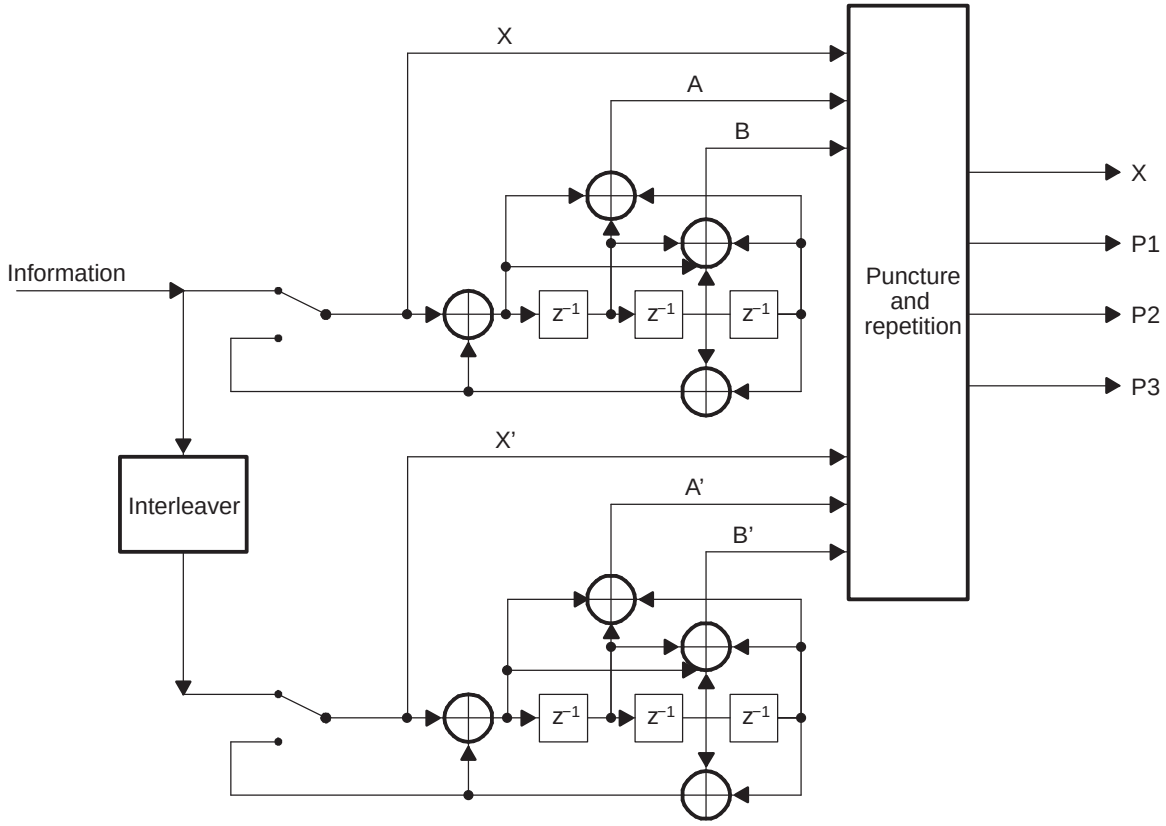
The TCP2 provides:

- High performance:
  - Very-low-processing delay because of the highly paralleled architecture allowing 8 iterations of a 2 Mbps 3GPP channel to be decoded in less than 1.2 ms and an IS2000 channel in less than 1.2 ms.
  - Processing delay can be further reduced by enabling a stopping criteria algorithm while achieving optimal BER performance.
  - TCP2 and the DSP can run full speed in parallel.
- System cost optimization:
  - Reduces board space and power consumption by performing on-chip turbo-decoding.
  - Communication between the DSP and the TCP2 is performed through a high performance DMA engine, the enhanced DMA (EDMA3).
  - TCP2 uses its own optimized memories, reducing system memory overhead and yielding higher overall performance.
  - Increased programmability.
  - Power efficient and module power-saver capabilities.
- High flexibility to cope with standard evolutions:
  - Accepts all IS2000, 3GPP rates, and polynomials.
  - Accepts any frame length from 40 (3GPP minimum frame size) up to 20730 for standalone processing. Frame sizes greater than 20730 can be processed by breaking them up into smaller subframes for processing in shared processing mode.
  - Supports all interleaver combinations via interleaver table.
  - Frees-up DSP resources.
- Improvements over TCP:
  - Standalone mode frame length increased from 5114 to 20730.
  - Code rates 1/2,1/3,1/4 and 1/5 (other rates via de-puncturing may be achieved).
  - Prolog reduction.
  - Cyclic redundancy check (CRC) stopping criteria.
  - Channel re-encoding.
  - Max-Log Maximum a Posteriori (MAP) option added to TCP2 (Max*-Log MAP still available).
  - Input sign programmable.
  - Debug mode added to allow pausing after each MAP.
  - Decision ordering programmable as MSB first or LSB first.
  - Extrinsic scaling added for Max-Log MAP.

2 Introduction

Encoding is done as shown in Figure 1. The 3GPP and IS2000 turbo encoders employ two recursive, systematic, convolutional (RSC) encoders connected in parallel, with an interleaver (the turbo interleaver) preceding the second recursive convolutional encoder. The two recursive convolutional codes are called the constituent encoders of the turbo code and have a constraint length K = 4.

Figure 1. 3GPP and IS2000 Turbo-Encoder Block Diagram



Switches in upper position for information bits and in lower position for tail bits

The outputs of the constituent encoders are punctured and repeated (F denotes the frame size, X and X' are systematic data, A, B, A', and B' are parity data, X', A', and B' are the interleaved versions of X, A, and B data):

- Data rate 1/2 ($2 \times F$ bits):
  $X_0A_0X_1A'_1X_2A_2X_3A'_3$…
- Data rate 1/3 ($3 \times F$ bits):
  $X_0A_0A'_0X_1A_1A'_1$ $X_2A_2A'_2$ $X_3A_3A'_3$ …
- Date rate 1/4 ($4 \times F$ bits):
  $X_0A_0B_0B'_0X_1A_1A'_1B'_1X_2A_2B_2B'_2X_3A_3A'_3B'_3$ …

For the tail bits, the sequence is:

- IS2000 tail rate 1/2 and 3GPP tail rate 1/3: 12 bits
  $X_FA_FX_{F+1}A_{F+1}X_{F+2}A_{F+2}X'_FA'_FX'_{F+1}A'_{F+1}X'_{F+2}A'_{F+2}$
- IS2000 tail rate 1/3: 18 bits (systematic bit repeated twice)
  $X_FX_FA_FX_{F+1}X_{F+1}A_{F+1}X_{F+2}X_{F+2}A_{F+2}X'_FX'_FA'_FX'_{F+1}X'_{F+1}A'_{F+1}X'_{F+2}X'_{F+2}A'_{F+2}$
- IS2000 tail rate 1/4: 24 bits (systematic bit repeated twice)
  $X_FX_FA_FB_F$ $X_{F+1}X_{F+1}A_{F+1}B_{F+1}$ $X_{F+2}X_{F+2}A_{F+2}B_{F+2}X'_FX'_FA'_FB'_FX'_{F+1}X'_{F+1}A'_{F+1}B'_{F+1}$ X'_{F+2} $X'_{F+2}A'_{F+2}B'_{F+2}$

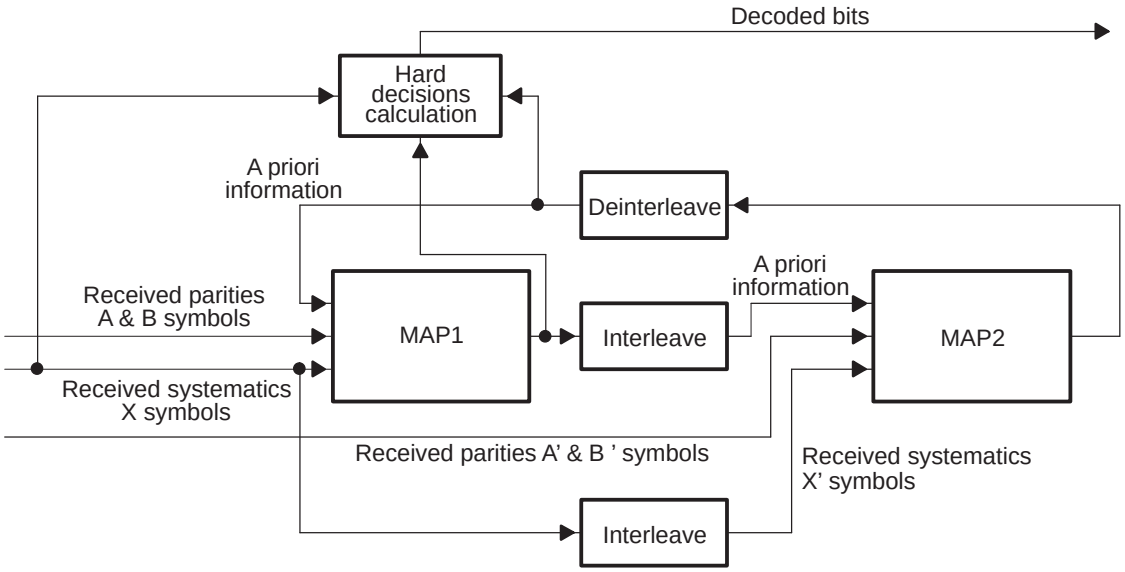
  The decoding process is an iterative algorithm based on simple decoders individually matched to the RSC codes. A generic 3GPP and IS2000 turbo decoder is shown in Figure 2.

  Each decoder sends *a posteriori* likelihood estimates of the decoded bits to the other decoder, and

uses the corresponding estimates from the other decoder as *a priori* likelihood. The *a priori* information is seen as beforehand knowledge, meaning that some messages are more likely to occur than others. *A posteriori* information adds to the *a priori* information the knowledge gained by the decoding.

The uncoded information bits (corrupted by the noisy channel) are available to each decoder to initialize the *a priori* likelihoods. The decoders use the Maximum a Posteriori (MAP) bitwise decoding algorithm that requires the same number of states as the well known Viterbi algorithm. The turbo decoder iterates between the outputs of the two constituent decoders until it reaches satisfactory convergence. The final output is a hard-quantized version of the likelihood estimates of the decoders.

Figure 2. 3GPP and IS2000 Turbo-Decoder Block Diagram



3 Overview

The DSP controls the operation of the TCP2 (Figure 3) using memory-mapped registers. The DSP typically sends and receives data using synchronized EDMA3 transfers through the 64-bit EDMA3 bus. The TCP2 sends two synchronization events to the EDMA3: a receive event (TCPREVT) and a transmit event (TCPXEVT).

The processing unit can implement the Max*-Log-MAP or Max-Log-MAP approximations of the BCJR algorithm and is selected with the E_MAX_STAR bit of the TCPIC3 register. (See L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate",. *IEEE Trans. Inform.Theory*, vol. IT.20, pp. 284.287, Mar. 1974 and P. Robertson, E. Villebrun, and P. Hoeher, "A comparison of optimal and sub-optimal MAP decoding algorithms operating in the log domain", in Proc. 1996 *IEEE Int. Conf. on Communications* (Seattle, WA), June 1995, vol. 2, pp. 1009-1013.)

The TCP2 has two fundamental modes: standalone (SA) and shared processing (SP).

In SA mode, the TCP2 iterates a given number of times and outputs hard decisions. In SP mode, the TCP2 executes a single MAP decode and outputs extrinsic information (soft information). SA mode is typically used for frame sizes up to 20730. SP mode must be used for frames strictly larger than 20730. Table 1 describes which mode to use depending on the frame size.

The TCP2 input data corresponds to channel log-likelihood ratios scaled on 6 bits, while the TCP2 output data to hard-decisions (SA mode) or extrinsics (SP mode) scaled on 7 bits.

Figure 3. TCP2 Block Diagram

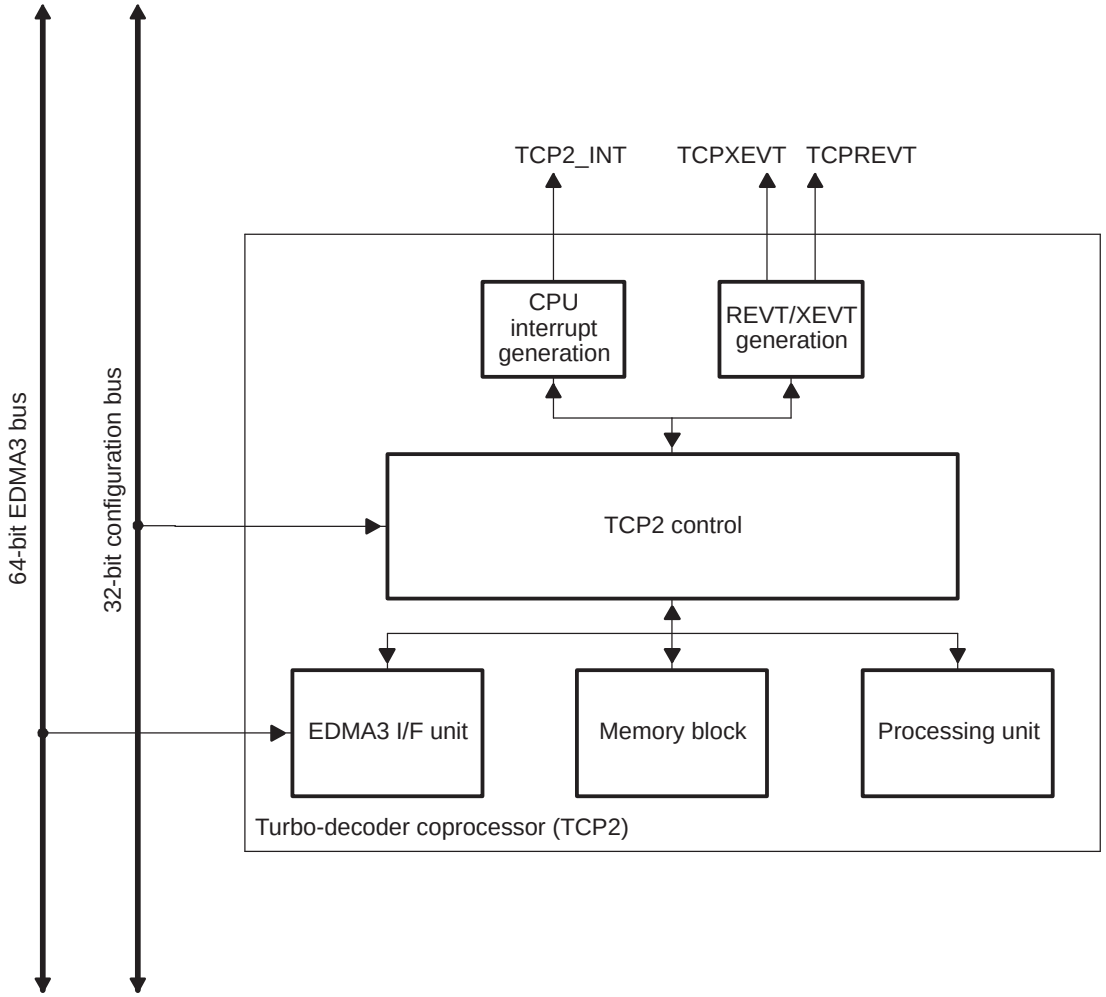


Table 1. Frame Sizes for Standalone (SA) Mode and Shared-Processing (SP) Mode

Frame Size (F)	TCP2 Mode
40 ⇐ F ⇐ 20730	Standalone mode
F > 20730	Shared processing

4 Standalone (SA) Mode

In standalone (SA) mode, the DSP sends the systematic and parity data, and the interleaver table. The TCP2 then works independently of the DSP (standalone), iterates a defined maximum number of times, and outputs hard decision data. In this mode, minimum DSP processing is required. A stopping criteria can be enabled to reduce the processing delay (see Section 4.3). Figure 4 shows the SA mode.

The standalone mode is used for frames in which the turbo interleaver length is less than or equal to 20730. In this mode, the systematic, parities, extrinsics, and turbo interleaver data fit within the TCP2 memory, and several iterations of decoding are run within the coprocessor without any DSP intervention.

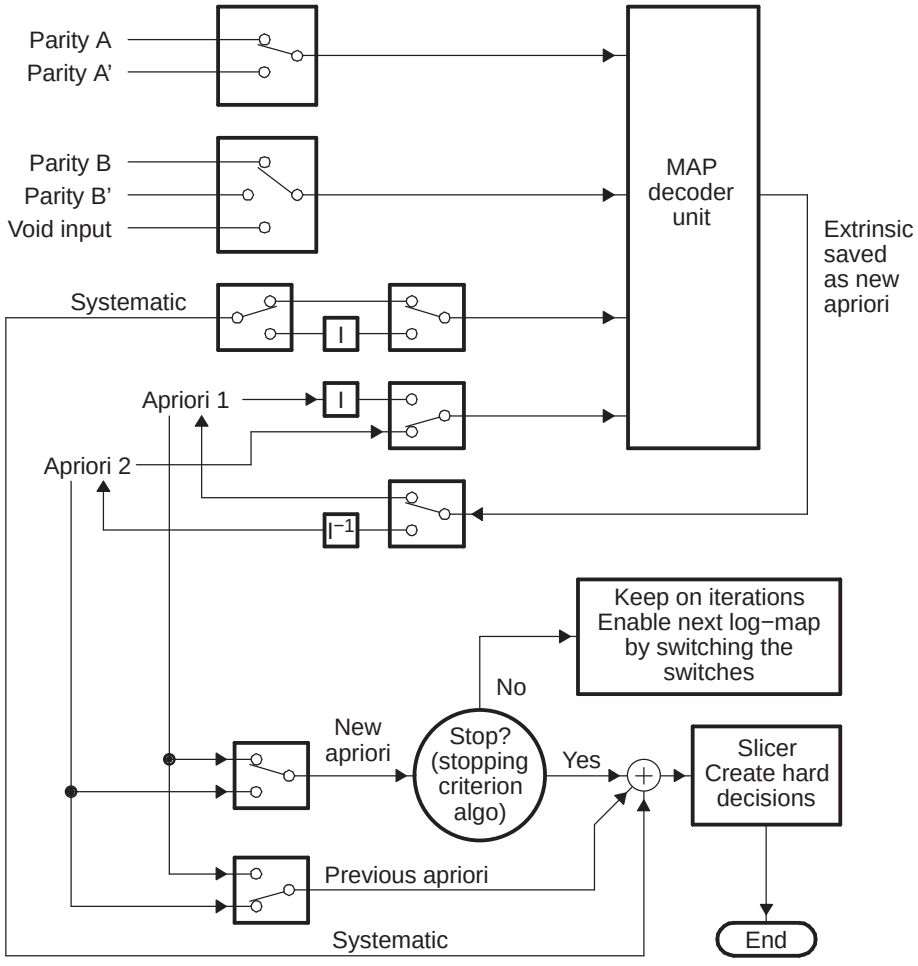
The DSP sets up the EDMA3 to send the systematic and parity data, and the interleaver table (optional). The TCP2 then works independently of the DSP.

One iteration of turbo decoding consists of 2 MAPs processing, the first MAP with the initial switch position (as shown in Figure 4), the second MAP with the other position of the switch. After each MAP, a stopping test can be performed based on the following methods. These tests are user configurable.

- Comparing the extrinsic SNR estimate to a SNR threshold (user defined)
- CRC pattern match
- Max iterations

When starting a decoding, you must supply a maximum number of iterations and optionally an SNR threshold ratio (or CRC) for the stopping test. If the stopping test is positive or the maximum number of iterations is reached, the decoding stops, the hard decisions are computed (from both extrinsic and systematic data), and then the coprocessor notifies EDMA3 that the processing is complete. In Figure 4, switch positions are for MAP0 and opposite positions are for MAP1.

Figure 4. Standalone (SA) Mode Block Diagram



4.1 Input Data Format

4.1.1 Systematic and Parity Data

Symbols (data) have to be quantized on 6 bits as (4,2) bit numbers, that is, SIII.FF (where S = sign bit, I = integer bit, F = fractional bit). Depending on the rate, Figure 6 through Figure 16 show how data must be organized in the DSP memory to conform to a rate that is 1/5 of the input data stream, which TCP2 requires. The base address must be double-word aligned. For big-endian configuration, see the TCP2 endian register (TCPEND) in Section 6.22. Also note that interleaved parities must be de-interleaved prior to being sent to TCP2.

Figure 5. Systematic/Parity Data for Rates 1/2, 1/3, 1/4, 1/5, and 3/4

63:62	61:56	55:50	49:44	43:38	37:32	31:30	29:24	23:18	17:12	11:6	5:0
RSVD	SP9	SP8	SP7	SP6	SP5	RSVD	SP4	SP3	SP2	SP1	SP0

Figure 6. EN = 1 (Little-Endian Mode) Rate = 1/2

Word N + 1					Word N				
SP9 0	SP8 A1'	SP7 0	SP6 0	SP5 X1	SP4 0	SP3 0	SP3 0	SP3 0	SP0 X0
Word N + 3					Word N + 2				
SP9 0	SP8 A3'	SP7 0	SP6 0	SP5 X3	SP4 0	SP3 0	SP3 0	SP3 0	SP0 X2

Figure 7. EN = 0 (Big-Endian Mode) Rate = 1/2

Word N					Word N + 1				
SP4 0	SP3 0	SP2 0	SP1 A0	SP0 X0	SP9 0	SP8 A1'	SP7 0	SP6 0	SP5 X1
Word N + 2					Word N + 3				
SP4 0	SP3 0	SP2 0	SP1 A2	SP0 X2	SP9 0	SP8 A3'	SP7 0	SP6 0	SP5 X3

Figure 8. EN = 1 (Little-Endian Mode) Rate = 1/3

Word N + 1					Word N				
SP9 0	SP8 A1'	SP7 0	SP6 A1	SP5 X1	SP4 0	SP3 A0'	SP2 0	SP1 A0	SP0 X0
Word N + 3					Word N + 2				
SP9 0	SP8 A3'	SP7 0	SP6 A3	SP5 X3	SP4 0	SP3 A2'	SP2 0	SP1 A2	SP0 X2

Figure 9. EN = 0 (Big-Endian Mode) Rate = 1/3

Word N					Word N + 1				
SP4 0	SP3 A0'	SP2 0	SP1 A0	SP0 X0	SP9 0	SP8 A1'	SP7 0	SP6 A1	SP5 X1
Word N + 2					Word N + 3				
SP4 0	SP3 A2'	SP2 0	SP1 A2	SP0 X2	SP9 0	SP8 A3'	SP7 0	SP6 A3	SP5 X3

Figure 10. EN = 1 (Little-Endian Mode) Rate = 1/4

Word N + 1					Word N				
SP9 B1'	SP8 A1'	SP7 0	SP6 A1	SP5 X1	SP4 B0'	SP3 0	SP2 B0	SP1 A0	SP0 X0
Word N + 3					Word N + 2				
SP9 B3'	SP8 A3'	SP7 0	SP6 A3	SP5 X3	SP4 B2'	SP3 0	SP2 B2	SP1 A2	SP0 X2

Figure 11. EN = 0 (Big-Endian Mode) Rate = 1/4

Word N					Word N + 1				
SP4 B0'	SP3 0	SP2 B0	SP1 A0	SP0 X0	SP9 B1'	SP8 A1'	SP7 0	SP6 A1	SP5 X1
Word N + 2					Word N + 3				
SP4 B2'	SP3 0	SP2 B2	SP1 A2	SP0 X2	SP9 B3'	SP8 A3'	SP7 0	SP6 A3	SP5 X3

Figure 12. EN = 1 (Little-Endian Mode) Rate = 1/5

Word N + 1					Word N				
SP9 B1'	SP8 A1'	SP7 B1	SP6 A1	SP5 X1	SP4 B0'	SP3 A0'	SP2 B0	SP1 A0	SP0 X0
Word N + 3					Word N + 2				
SP9 B3'	SP8 A3'	SP7 B3'	SP6 A3	SP5 X3	SP4 B2'	SP3 A2'	SP2 B2	SP1 A2	SP0 X2

Figure 13. EN = 0 (Big-Endian Mode) Rate = 1/5

Word N					Word N + 1				
SP4 B0'	SP3 A0'	SP2 B0	SP1 A0	SP0 X0	SP9 B1'	SP8 A1'	SP7 B1	SP6 A1	SP5 X1
Word N + 2					Word N + 3				
SP4 B2'	SP3 A2	SP2 B2	SP1 A2	SP0 X2	SP9 B3'	SP8 A3'	SP7 B3	SP6 A3	SP5 X3

Figure 14. EN = 1 (Little-Endian Mode) Rate = 3/4

Word N + 1					Word N				
SP9 0	SP8 0	SP7 0	SP6 0	SP5 X1	SP4 0	SP3 0	SP2 0	SP1 A0	SP0 X0
Word N + 3					Word N + 2				
SP9 0	SP8 A3'	SP7 0	SP6 0	SP5 X3	SP4 0	SP3 0	SP2 0	SP1 0	SP0 X2
Word N + 5					Word N + 4				
SP9 0	SP8 0	SP7 0	SP6 0	SP5 X3	SP4 0	SP3 0	SP2 0	SP1 0	SP0 X2

Figure 15. Rate 3/4 EN = 0 (Big-Endian Mode) Rate = 3/4

Word N					Word N + 1				
SP4 0	SP3 0	SP2 0	SP1 A0	SP0 X0	SP9 0	SP8 0	SP7 0	SP6 0	SP5 X1
Word N + 2					Word N + 3				
SP4 0	SP3 0	SP2 0	SP1 0	SP0 X2	SP9 0	SP8 A3'	SP7 0	SP6 0	SP5 X3
Word N + 4					Word N + 5				
SP4 0	SP3 0	SP2 0	SP1 0	SP0 X4	SP9 0	SP8 0	SP7 0	SP6 0	SP5 X5

4.1.2 Interleaver Indexes

Each index is a 15-bit value being effectively saved as 16 bits right-justified. Given an index *j*, an interleaver table *t*, and a data *x*, the interleaved data *x* is given as *x*' = x[*t*(*j*)]. Table 2 shows how data must be organized in the memory. The base address must be double-word aligned. For big-endian configurations, see the TCP2 endian register (TCPEND) in Section 6.22.

Table 2. Interleaver Data

Little_big_endian	Endian_intr	Description (MSB to LSB)
0	0	1,0,3,2 ⇒ 3,2,1,0 (halfword)
0	1	0,1,2,3 ⇒ 3,2,1,0 (halfword)
1	0	Endianness manager has no effect 3,2,1,0 ⇒ 3,2,1,0 (halfword)
1	1	Endianness manager has no effect 3,2,1,0 ⇒ 3,2,1,0 (halfword)

4.2 Output Decision Data Format

Hard decisions for TCP2 are 32-bit word-packed. The bit ordering within the 32-bit hard-decision word is programmable, such that the oldest bit can be either in the MSB or the LSB position. Their destination storage base address must be double-word aligned. Moreover, the buffer length must contain an even number of words.

4.3 Stopping Criteria

The turbo decoder has an iterative structure, and the number of iterations that are performed for each frame is either a deterministic number or it depends on a test performed on the turbo decoder output after each iteration. In the first case, you decide how many iterations should be performed prior to decoding a frame. In the second case, the turbo decoder performs tests after each iteration to determine whether the iterative process should continue. In this case, the boundary conditions are programmed (for example, the minimum and the maximum number of iterations that should be performed). The tests performed are the SNR stopping criterion and cyclic redundancy check (CRC) iterations passed.

This SNR stopping criterion on TCP2 can be used by setting the SNR threshold from 1 to 100 (0 disables the SNR threshold check). The stopping criteria is met and a TCPREVT is generated when the SNR threshold is met, the minimum iterations have been processed, and sufficient CRC iterations have passed, if CRC is enabled. This indicates that decisions are ready for the EDMA3 to access.

Larger thresholds improve bit-error rate (BER) performance, but require more iterations. Smaller thresholds require fewer iterations, but may yield poorer BER performance. The actual number of iterations run can be read from the output parameters.

The CRC-based stopping criterion can be used by setting the CRC polynomial length (CRCLEN) and the number of CRC iterations required to pass CRCITERPASS. After each iteration, hard decisions are computed and a CRC is performed. The CRC polynomial is a programmable 32-bit number. To avoid situations where a CRC test passes for a very noisy frame of data, the hard decisions need to pass the CRC test for a number of consecutive iterations, which is user-defined via the CRCITERPASS bit field.

4.4 Stopping Test Unit

Turbo decoders are iterative decoders. Each iteration consists of two MAP decodes except the last iteration that executes only the first MAP decode. The turbo decoder can iterate up to 32 iterations. The decoder will continue to iterate until one of the following conditions occur: meet parameter conditions, CRC passed, or SNR threshold passed.

4.4.1 SNR Threshold Termination

The stopping criteria algorithm generates the first two moments of the extrinsics, generates an SNR ratio, and compares the ratio with a threshold. If the calculated ratio exceeds the threshold, then the decoder has found an optimum solution. The decoder can then stop executing any further iterations. The calculated SNR ratio is generated after each MAP process. The threshold is a user input and can range from 0 to 100. Larger thresholds give better results but require more iterations. Smaller thresholds require fewer iterations and give can give poorer results. Setting the threshold to 0 disables the stopping criteria algorithm.

The stopping criteria contains two parts. The first part executes on each extrinsic value. The sum of the extrinsics and the sum of the extrinsics squared are calculated. The second part is executed once at the end of each MAP block. The first moment is squared and multiplied by the sum of 1 plus the inverse of the threshold. The second moment is multiplied by the number of symbols per frame. The two results are compared. If the result is positive, then the stopping criteria has been met.

The turbo decoder will generate a block of extrinsics after each MAP decode. The SNR stopping criteria block calculates the mean and the variance for this block. It will divide the two and compare the result with the snr_threshold. If the result is greater then the snr_threshold for two consecutive MAP decodes, then the decoder will stop executing. The DSP sets the snr_threshold parameter. The SNR stopping criteria can be turned off with a value of 0. Enabled values for snr_threshold range from 1 to 100. A value of 100 gives the best BER performance at a cost of the most iterations executed, and a value of 1 gives the worst BER performance at a cost of the fewest iterations. Recommended setting for this parameter is 100.

4.4.2 CRC Termination

A frame of data is sent through a CRC block which appends crc_length number of bits to the frame. This frame is encoded by the turbo encoder. The polynomial for the CRC check is defined with the crc_poly parameter. The turbo decoder will generate hard decision bits after each non-interleaved MAP decode. These bits are processed by the CRC block within the decoder. If the last crc_length bits match the CRC pattern, then the CRC check has passed. The turbo decoder will stop executing after CRCITERPASS number of consecutive CRC passes as programmed in TCPIC4.

The coefficients and the size of the CRC polynomial are programmable. The size of the polynomial is defined with the parameter crc_length and can be set from 0 to 32 bits. A value of 0 disables the CRC check, values between 1 and 32 enable the CRC check. The CRC polynomial is defined with the crc_poly parameter. The CRC unit will not be enabled until the decoder iteration count is equal or greater than the min_iter parameter. The turbo decoder will generate hard decisions after each non-interleaved MAP decode. These bits are processed by the CRC block within the decoder. If the last set of frame bits match the CRC pattern, then the CRC check has passed. The turbo decoder must pass a number of consecutive iterations to terminate before max_iter. The number of consecutive iterations passed is defined with the crc_iter_pass parameter. The crc_iter_pass parameter can be set from 0 to 31, a zero is equal to 1 iteration. The dec_pass output parameter will be set to a 1 if the decoder terminated due to a passing CRC.

During the sub-block execution, up to 256 sets of data will be stored in a double buffered RAM whose size is 265x7x2. Two bits each will be stored for x0, p0, and p1. One bit is the sign bit and the other bit is set if the symbol is equal to a zero. These 6 bits will be used for re-encoding. The seventh bit will be the hard decision bit. This bit is the sign of the following summation: (x+a+w).

The CRC will process one sub-block at time using the data stored from the previous sub-block. The decision bit will be used by a CRC block. After all sub-blocks have been processed, the CRC bits in the CRC block are checked and compared with the last crc_length bits of the frame. If they all match, then the CRC passes.

4.4.3 Parameter Termination

The parameters min_iter and max_iter need to be set prior to decode. The decoder must execute min_iter number of iterations. This parameter can be set from 0 to 31. The decoder will stop executing when the iteration count equals max_iter. Max_iter can be set from 0 to 31 and must be equal or greater than min_iter. A zero for max is equal to 32 iterations. A zero for min is equal to 1 iteration.

4.4.3.1 Maximum Iterations

Turbo decoders execute the MAP decoder twice per iteration. One execution is for non-interleaved data and the other execution is for interleaved data. This parameter sets the maximum number of decoder iterations for each block of data. Valid sizes are 0 to 31. If either the CRC passed or the SNR stopping criteria threshold has been exceeded, then the decoder will stop early. The last iteration will only process the MAP decoder for the non-interleaved data.

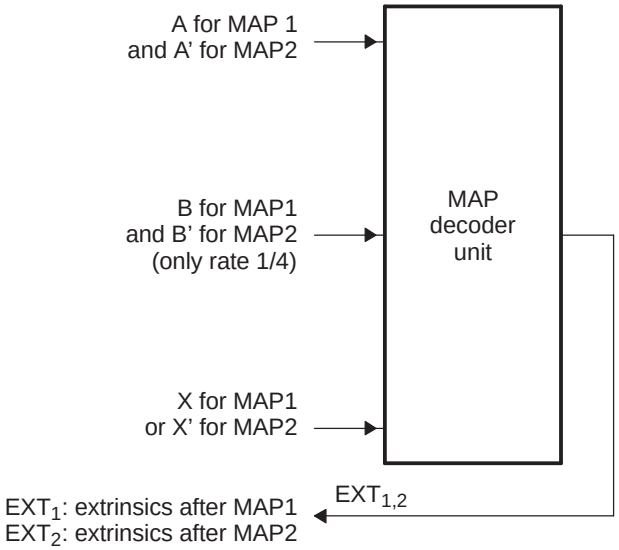
4.4.3.2 Minimum Iterations

Turbo decoders execute the MAP decoder twice per iteration. One execution is for non-interleaved data and the other execution is for interleaved data. This parameter sets the minimum number of decoder iterations for each block of data. Valid sizes are 0 to 31 and the min_iter must be less than or equal to the max_iter. The CRC unit will not be enabled until the decoder iteration count is equal or greater than the min_iter parameter. The turbo decoder will not process the CRC, re-encode, or write to the output RAM until the minimum number of iterations has been reached.

5 Shared-Processing (SP) Mode

In shared-processing (SP) mode, the DSP sends systematic and parity data, and a priori data. The TCP performs one single MAP decode and outputs extrinsic data. A priori data for MAP1 is obtained by de-interleaving the extrinsic data from the previous MAP2, and a priori data for MAP2 is obtained by interleaving the extrinsics data from the previous MAP1. An overview of the SP mode is shown in Figure 16. Note that the systematic and parity data to be sent to the TCP has to be demultiplexed from the original flow described in Section 5.1. The DSP must perform the input data demultiplexing, interleaving, deinterleaving operations, hard decision calculation, and any stopping criteria algorithm.

Figure 16. Shared-Processing (SP) Mode Block Diagram



The shared-processing mode allows the DSP/TCP2 system to support frames strictly larger than 20730. The DSP breaks the large frame into 2 or more smaller frames of 20480 or less. Each frame is called a subframe. The size of all the subframes (except the last subframe) must be divisible by 256. The DSP breaks the large frame into several sub-frames following the process shown below. The first subframe does not have a header section and its tail section is equal to the prolog size. The middle subframe header and tail section sizes are each equal to the prolog size. The last subframe header size is equal to the prolog size and the tail section is equal to the tail size. The prolog size must be integer divisible by 8. The sub-frames reliability portions are sequenced in order to create the full frame.

1. The first subframe (required) will have a opmode of 1. The middle subframe(s) (optional) have a opmode of 2. The last subframe (required) will have a opmode of 3.
2. The size of all the subframes (except the last subframe) must be integer divisible by 256 and Max sub frame size = 20480 = 80*256.
3. The first and middle subframes should have the same size. The last subframe should be approximately the same size as the other subframes.
4. The last subframe size must be at least 129.

Figure 17. Subframe Equations

$$Num_{Subframe} = CEIL\left(\frac{Size_{Block}}{Size_{MAX\_Subframe}}\right)$$

$$Size_{Subframe} = CEIL\left(\frac{Size_{Block}}{256 \times Num_{Subframe}}\right) \times 256$$

$$while\left(Size_{Block} > Size_{MAX\_Subsystem}\right) \{$$

$$Size_{Block} = Size_{Block} - Size_{Subframe}$$

$$\}$$

$$if(Size_{Block} > 128)$$

$$Size_{Last\_Subframe} = Size_{Block}$$

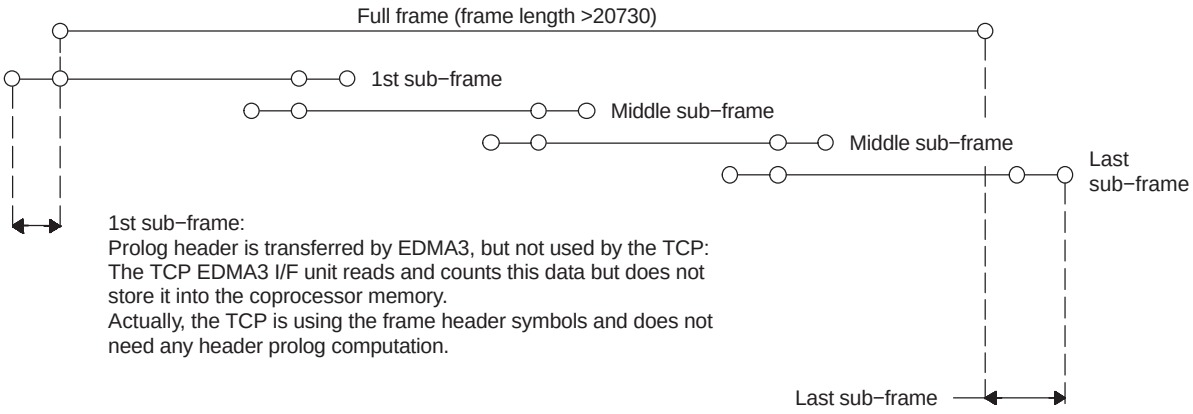
$$if(Size_{Block} \leq 128) \{$$

$$Num_{Subframe} = Num_{Subframe} - 1$$

$$Size_{Last\_Subframe} = Size_{Block} + Size_{MAX\_Subframe}$$

$$\}$$

Figure 18. Frame Process



Each sub-frame is independent of each other. There are three types of sub-frames. The first sub-frame starts the trellis from the zero state. The last sub-frame ends the trellis from a known state. The remaining middle subframes do not start or end from a known state.

The EDMA3 transfers ACNT*BCNT number of bytes in A-Sync Mode and ACNT*BCNT*CCNT number of bytes in AB-Sync Mode. The total number of bytes for both modes should be a multiple of 8. Also, the starting address of the first sub-frame that the EDMA3 will transfer needs to be memory-mapped.

In the shared processing mode:

- Prolog length must be multiples of 8
- Starting address for reading extrinsic RAM must be:
  RAM base address + middle and last subframes prolog length
- CRC is turned off
- SNR is turned off
- Prolog reduction is turned off
- Extrinsic scaling is turned off

The turbo decoding of the full frame is performed in several steps as described below:

- The EDMA3 sends the input buffers for one sub-frame (the MAP0 inputs are described in Figure 19).
- The TCP2 performs the MAP0 for the current sub-frame.
- The EDMA3 reads the MAP output (extrinsic) of the current sub-frame and writes it into the DSP memory.

The steps for the MAP0 process are repeated for all the other sub-frames.

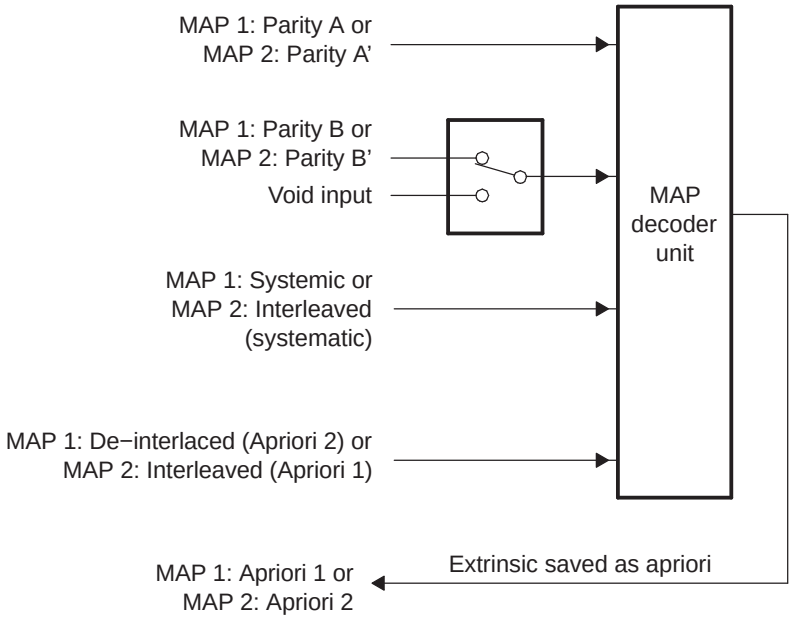
Once all the sub-frames MAP0 have been computed, the full MAP0 extrinsic (= apriori 1) is then available. This allows the DSP to interleave the extrinsic output 1 to prepare the next MAP (= MAP1). Once this interleaving is done, the same process is applied, in MAP1 configuration:

- The EDMA3 sends the input buffers for one sub-frame (the MAP0 inputs are described in Figure 19).
- The TCP2 performs the MAP1 for the current sub-frame
- The EDMA3 reads the MAP output (extrinsic) of the current sub-frame and writes it into the DSP memory.

The steps for the MAP1 process are repeated for all the other sub-frames.

Once all the sub-frames MAP1 have been computed, the full extrinsic (=apriori 2) is then available. This allows the DSP to de-interleave the extrinsic output 2 to prepare the next MAP (=MAP0). Once this de-interleaving is done, the same process is applied, in MAP0 configuration. Steps 1-4 are then repeated for all iterations. The DSP is in charge of any stopping criteria algorithm implementation and computing the final hard decisions. Figure 19 shows a description of the TCP2 processing unit functional block diagram in shared processing mode.

Figure 19. TCP2 Shared Processing Block Diagram



5.1 Input Data Format

5.1.1 Systematic and Parity Data

The original systematic and parity data is organized as described in Section 4.1.1. The DSP has to split the data for MAP0 and MAP1 as shown in the following figures. The base address must be double-word aligned. Interleaved systematic data (X.) must be calculated by the DSP given the interleaver table (see Section 4.1.2). For big-endian configuration, see Section 7.

Figure 20. Systematic/Parity Data for Rates 1/2, 1/3, 1/4, 1/5, and 3/4

63:62	61:56	55:50	49:44	43:38	37:32	31:30	29:24	23:18	17:12	11:6	5:0
RSVD	SP9	SP8	SP7	SP6	SP5	RSVD	SP4	SP3	SP2	SP1	SP0

Figure 21. EN = 1 (Little-Endian Mode) Rate = 1/2

Word N + 1					Word N				
SP9 0	SP8 A1'	SP7 0	SP6 0	SP5 X1	SP4 0	SP3 0	SP3 0	SP3 0	SP0 X0
Word N + 3					Word N + 2				
SP9 0	SP8 A3'	SP7 0	SP6 0	SP5 X3	SP4 0	SP3 0	SP3 0	SP3 0	SP0 X2

Figure 22. EN = 0 (Big-Endian Mode) Rate = 1/2

Word N					Word N + 1				
SP4 0	SP3 0	SP2 0	SP1 A0	SP0 X0	SP9 0	SP8 A1'	SP7 0	SP6 0	SP5 X1
Word N + 2					Word N + 3				
SP4 0	SP3 0	SP2 0	SP1 A2	SP0 X2	SP9 0	SP8 A3'	SP7 0	SP6 0	SP5 X3

Figure 23. EN = 1 (Little-Endian Mode) Rate = 1/3

Word N + 1					Word N				
SP9 0	SP8 A1'	SP7 0	SP6 A1	SP5 X1	SP4 0	SP3 A0'	SP2 0	SP1 A0	SP0 X0
Word N + 3					Word N + 2				
SP9 0	SP8 A3'	SP7 0	SP6 A3	SP5 X3	SP4 0	SP3 A2'	SP2 0	SP1 A2	SP0 X2

Figure 24. EN = 0 (Big-Endian Mode) Rate = 1/3

Word N					Word N + 1				
SP4 0	SP3 A0'	SP2 0	SP1 A0	SP0 X0	SP9 0	SP8 A1'	SP7 0	SP6 A1	SP5 X1
Word N + 2					Word N + 3				
SP4 0	SP3 A2'	SP2 0	SP1 A2	SP0 X2	SP9 0	SP8 A3'	SP7 0	SP6 A3	SP5 X3

Figure 25. EN = 1 (Little-Endian Mode) Rate = 1/4

Word N + 1					Word N				
SP9 B1'	SP8 A1'	SP7 0	SP6 A1	SP5 X1	SP4 B0'	SP3 0	SP2 B0	SP1 A0	SP0 X0
Word N + 3					Word N + 2				
SP9 B3'	SP8 A3'	SP7 0	SP6 A3	SP5 X3	SP4 B2'	SP3 0	SP2 B2	SP1 A2	SP0 X2

Figure 26. EN = 0 (Big-Endian Mode) Rate = 1/4

Word N					Word N + 1				
SP4 B0'	SP3 0	SP2 B0	SP1 A0	SP0 X0	SP9 B1'	SP8 A1'	SP7 0	SP6 A1	SP5 X1
Word N + 2					Word N + 3				
SP4 B2'	SP3 0	SP2 B2	SP1 A2	SP0 X2	SP9 B3'	SP8 A3'	SP7 0	SP6 A3	SP5 X3

Figure 27. EN = 1 (Little-Endian Mode) Rate = 1/5

Word N + 1					Word N				
SP9 B1'	SP8 A1'	SP7 B1	SP6 A1	SP5 X1	SP4 B0'	SP3 A0'	SP2 B0	SP1 A0	SP0 X0
Word N + 3					Word N + 2				
SP9 B3'	SP8 A3'	SP7 B3'	SP6 A3	SP5 X3	SP4 B2'	SP3 A2'	SP2 B2	SP1 A2	SP0 X2

Figure 28. EN = 0 (Big-Endian Mode) Rate = 1/5

Word N					Word N + 1				
SP4 B0'	SP3 A0'	SP2 B0	SP1 A0	SP0 X0	SP9 B1'	SP8 A1'	SP7 B1	SP6 A1	SP5 X1
Word N + 2					Word N + 3				
SP4 B2'	SP3 A2	SP2 B2	SP1 A2	SP0 X2	SP9 B3'	SP8 A3'	SP7 B3	SP6 A3	SP5 X3

Figure 29. EN = 1 (Little-Endian Mode) Rate = 3/4

Word N + 1					Word N				
SP9 0	SP8 0	SP7 0	SP6 0	SP5 X1	SP4 0	SP3 0	SP2 0	SP1 A0	SP0 X0
Word N + 3					Word N + 2				
SP9 0	SP8 A3'	SP7 0	SP6 0	SP5 X3	SP4 0	SP3 0	SP2 0	SP1 0	SP0 X2
Word N + 5					Word N + 4				
SP9 0	SP8 0	SP7 0	SP6 0	SP5 X3	SP4 0	SP3 0	SP2 0	SP1 0	SP0 X2

Figure 30. Rate 3/4 EN = 0 (Big-Endian Mode) Rate = 3/4

Word N					Word N + 1				
SP4 0	SP3 0	SP2 0	SP1 A0	SP0 X0	SP9 0	SP8 0	SP7 0	SP6 0	SP5 X1
Word N + 2					Word N + 3				
SP4 0	SP3 0	SP2 0	SP1 0	SP0 X2	SP9 0	SP8 A3'	SP7 0	SP6 0	SP5 X3
Word N + 4					Word N + 5				
SP4 0	SP3 0	SP2 0	SP1 0	SP0 X4	SP9 0	SP8 0	SP7 0	SP6 0	SP5 X5

5.1.2 A Priori Data

A priori data for MAP0 and MAP1 must be organized as described in Figure 31 (the base address must be double-word aligned). For big-endian configuration, see Section 7.

Figure 31. A Priori Data

63:62	61:56	55:50	49:44	43:38	37:32	31:30	29:24	23:18	17:12	11:6	5:0
RSVD	AP4	AP3	AP2	AP1	AP0	RSVD	AP9	AP8	AP7	AP6	AP5

5.2 Output Data Format

The TCP2 delivers 32-bit word-packed extrinsic data. Each extrinsic is a (5,2) number; that is, SIIII.FF (where S = sign bit, I = integer, F = fractional bit) and is right-justified with 0 in the most-significant-bit position. Their destination storage base address must be double-word aligned. Moreover, the buffer length must contain an even number of words.

6 Registers

The TCP2 contains several memory-mapped registers accessible via the CPU, QDMA, and EDMA3. A peripheral-bus access is faster than an EDMA3-bus access for isolated accesses (typically when accessing control registers). EDMA3-bus accesses are intended to be used for EDMA3 transfers and are meant to provide maximum throughput to/from the TCP2.

The memory map is listed in Table 3, including all TCP2 memories (systematic and parity, interleaver, hard decisions, a priori, and extrinsic). All addresses provided are offset addresses. For the TCP2 base data address and TCP2 base control address, see the device-specific data manual.

Table 3. TCP2 Registers

TCP2 Data Offset Address	TCP2 Control Offset Address	Register/Memory Abbreviation	Name	See
	0x00000	TCPPID	TCP Peripheral Identification Register	Section 6.1
0x00000		TCPIC0	TCP Input Configuration Register 0	Section 6.2
0x00004		TCPIC1	TCP Input Configuration Register 1	Section 6.3
0x00008		TCPIC2	TCP Input Configuration Register 2	Section 6.4
0x0000C		TCPIC3	TCP Input Configuration Register 3	Section 6.5
0x00010		TCPIC4	TCP Input Configuration Register 4	Section 6.6
0x00014		TCPIC5	TCP Input Configuration Register 5	Section 6.7
0x00018		TCPIC6	TCP Input Configuration Register 6	Section 6.8
0x0001C		TCPIC7	TCP Input Configuration Register 7	Section 6.10
0x00020		TCPIC8	TCP Input Configuration Register 8	Section 6.11
0x00024		TCPIC9	TCP Input Configuration Register 9	Section 6.12
0x00028		TCPIC10	TCP Input Configuration Register 10	Section 6.13
0x0002C		TCPIC11	TCP Input Configuration Register 11	Section 6.14
0x00030		TCPIC12	TCP Input Configuration Register 12	Section 6.15
0x00034		TCPIC13	TCP Input Configuration Register 13	Section 6.16
0x00038		TCPIC14	TCP Input Configuration Register 14	Section 6.17
0x0003C		TCPIC15	TCP Input Configuration Register 15	Section 6.18
0x00040		TCPOUT0	TCP Output Parameters Register 0	Section 6.19
0x00044		TCPOUT1	TCP Output Parameters Register 1	Section 6.20
0x00048		TCPOUT2	TCP Output Parameters Register 2	Section 6.21
	0x0004C	TCPEXE	TCP Execute Register	Section 6.22
	0x00050	TCPEND	TCP Endianness Register	Section 6.23
	0x00060	TCPERR	TCP Error Register	Section 6.24
	0x00068	TCPSTAT	TCP Status Register	Section 6.25
	0x00070	TCPEMU	TCP Emulation Register	Section 6.26

Table 4. TCP2 RAMs

TCP2 Data Offset Address	Register/Memory Abbreviation	Name	Address Range	Length
0x10000	X0	Data/Sys and Parity Memory	0x10000-0x243FF	0x00014400
0x30000	W0	Extrinsic Mem 0	0x30000-0x351FF	0x00005100
0x40000	W1	Extrinsic Mem 1	0x40000-0x451FF	0x00005100
0x50000	I0	Interleaver Memory	0x50000-0x5a1FF	0x0000A200
0x60000	O0	Output/Decision Memory	0x60000-0x60a7F	0x00000A20
0x70000	S0	Scratch Pad Memory	0x70000-0x70aFF	0x000006E0
0x80000	T0	Beta State Memory	0x80000-0x80FFF	0x00000A00
0x90000	C0	CRC Memory	0x90000-0x90FFF	0x000001C0

Table 4. TCP2 RAMs (continued)

TCP2 Data Offset Address	Register/Memory Abbreviation	Name	Address Range	Length
0xA0000	B0	Beta Prolog Memory	0xa0000-0xa0FFF	0x00000280
0xB0000	A0	Alpha Prolog Memory	0xb0000-0xb0FFF	0x00000280

6.1 Peripheral Identification Register (PID)

The peripheral identification register (PID) is a constant register that contains the ID and ID revision number for that peripheral. The PID stores version information used to identify the peripheral. All bits within this register are read-only (writes have no effect) meaning that the values within this register should be hard-coded with the appropriate values and must not change from their reset state. The peripheral identification register (PID) is shown in Figure 32 and described in Table 5. TCPIC0 configures the TCP.

Figure 32. Peripheral Identification Register (PID)

31	24	23	16
Reserved		TYPE	
R-0		R-type	

15	8	7	0
CLASS		REV	
R-class		R-rev	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 5. Peripheral Identification Register (PID) Field Descriptions

Bit	Field	Value	Description
31-24	Reserved	0	Reserved. The reserved bit location is always read as 0. A value written to this field has no effect.
23-16	TYPE		Peripheral type. Identifies the type of the peripheral. Set to 0x02 by default.
15-8	CLASS		Peripheral class. Identifies the class. Set to 0x11 by default.
7-0	REV		Peripheral revision. Identifies the revision level of the specific instance of the peripheral. This value should begin at 0x01 and be incremented each time the design is revised.

6.2 TCP2 Input Configuration Register 0 (TCPIC0)

The TCP2 input configuration register 0 (TCPIC0) is shown in Figure 33 and described in Table 6. TCPIC0 configures the TCP.

Figure 33. TCP2 Input Configuration Register 0 (TCPIC0)

31	30	16
Rsvd	FL	
R/W-0	R/W-0	

15	14	13	12	11	10	8
Reserved	NUMSW	OUTF	INTER	Reserved	RATE	
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	

7	3	2	1	0
Reserved		OPMOD		Reserved
R/W-0		R/W-0		R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 6. TCP2 Input Configuration Register 0 (TCPIC0) Field Descriptions

Bit	Field	Value	Description
31	Reserved	0	Reserved. The reserved bit location is always read as 0. A value written to this field has no effect.
30-16	FL	40-20730	Frame length. Frame size (should not include tail symbols).
15	Reserved	0	Reserved. The reserved bit location is always read as 0. A value written to this field has no effect.
14	NUMSW		Number of slide window per sub-block
		0	Block size 3 128
		1	Block size >128
13	OUTF		Output parameters read flag (SA mode only, must be set to 0 for SP mode).
		0	No REVT generation. Output parameters are not read via EDMA3.
		1	REVT generation for output parameters for EDMA3 read.
12	INTER		Interleaver write flag. Only used in standalone mode.
		0	Interleaver table is not sent to the TCP2
		1	Interleaver table is sent to the TCP2
11	Reserved	0	Reserved. The reserved bit location is always read as 0. A value written to this field has no effect.
10-8	RATE		Code rate = 1/rate except for 0 = 3/4
7-3	Reserved	0	Reserved. The reserved bit location is always read as 0. A value written to this field has no effect.
2-1	OPMOD	0-3h	Operational mode. This two bit signal defines the processing mode for the TCP2. If the size of the frame is less than or equal to 20,730, then the TCP2 is in the more efficient standalone mode. If the size of the frame is greater than 20,730, then the TCP2 is in shared mode.
		00	Standalone mode
		01	Shared-processing mode first subframe
		10	Shared-processing mode middle subframe
		11	Shared-processing mode last
			Shared mode breaks the frame into smaller frames called subframes. The size of each subframe (except the last subframe) must be an integer multiple of 256. For subframe equations, see Figure 17.
0	Reserved	0	Reserved. The reserved bit location is always read as 0. A value written to this field has no effect.

In the above register, when the OPMOD field is 01b, it represents the first subframe, 10b represents middle subframes, and 11b represents the last subframe. The TCP2 needs to know which type of subframe it is processing so that it can correctly initialize the prolog sections.

6.3 TCP2 Input Configuration Register 1 (TCPIC1)

The TCP2 input configuration register 1 (TCPIC1) is shown in Figure 34 and described in Table 7. TCPIC1 configures the TCP.

Figure 34. TCP2 Input Configuration Register 1 (TCPIC1)

31 23	22 16	15 0
Reserved	R	Reserved
R/W-0	R/W-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

Table 7. TCP2 Input Configuration Register 1 (TCPIC1) Field Desccriptions

Bit	Field	Value	Description
31-23	Reserved	0	Reserved. The reserved bit location is always read as 0. A value written to this field has no effect.
22-16	R	39-127	Reliability length + 1: 7 bits (min = 39, max = 127).
14-0	Reserved	0	Reserved. The reserved bit location is always read as 0. A value written to this field has no effect.

6.4 TCP2 Input Configuration Register 2 (TCPIC2)

The TCP2 input configuration register 2 (TCPIC2) is shown in Figure 35 and described in Table 8. TCPIC2 configures the TCP.

Figure 35. TCP2 Input Configuration Register 2 (TCPIC2)

31 24	23 21	20 16
SNR	Reserved	MAXIT
R/W-0	R/W-0	R/W-0

15	14 8	7 6	5 0
Rsvd	NSB	Reserved	P
R/W-0	R/W-0	R/W-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

Table 8. TCP2 Input Configuration Register 2 (TCPIC2) Field Descriptions

Bit	Field	Value	Description
31-24	SNR	0-100	SNR threshold. SNR is used for stopping test (8 bits). The turbo decoding will stop as soon as the decoded signal SNR ratio > SNR threshold ratio. The maximum value is 100, giving a value of zero disables this test and the decoder will execute the number of iterations given in the above parameter.
		0	Disables ratio threshold
		1 to 100	Enables threshold ratio
23-21	Reserved	0	Reserved. The reserved bit location is always read as 0. A value written to this field has no effect.
20-16	MAXIT	0-31	Maximum number of iterations: 5 bits (0-31). 0 means 32 iterations.
		0	Sets MAXIT to 32.
15	Reserved	0	Reserved. The reserved bit location is always read as 0. A value written to this field has no effect.
14-8	NSB	0-81	Number of sub-blocks. For shared processing mode only.
7-6	Reserved	0	Reserved. The reserved bit location is always read as 0. A value written to this field has no effect.
5-0	P	4-48	Prolog length (from 4 to 48). Sliding window prolog length, maximum is 48, minimum is 4. In shared-processing mode, the prolog length must be a multiple of 8, due to EDMA3 transfers alignment.

6.5 TCP2 Input Configuration Register 3 (TCPIC3)

The TCP2 input configuration register 3 (TCPIC3) is shown in Figure 36 and described in Table 9. TCPIC3 informs the TCP2 on the EDMA3 data flow segmentation.

Figure 36. TCP2 Input Configuration Register 3 (TCPIC3)

31	16
Reserved	
R/W-0	

15	14	13	12	11	9	8
Reserved	OUT ORDER	Reserved	INPUT SIGN	Reserved		MINITER
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0		R/W-0

7	4	3	2	1	0
MINITER		Reserved		EPRORED	EXMASTR
R/W-0		R/W-0		R/W-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 9. TCP2 Input Configuration Register 3 (TCPIC3)

Bit	Field	Value	Description
31-15	Reserved	0	Reserved. The reserved bit location is always read as 0. A value written to this field has no effect.
14	OUTORDER		Output bit ordering.
		0	Output bit ordering from 0 to 31
		1	Output bit ordering from 31 to 0
13	Reserved	0	Reserved. The reserved bit location is always read as 0. A value written to this field has no effect.
12	INPUTSIGN		Multiply channel input data.
		0	Multiply channel input data by + 1
		1	Multiply channel input data by - 1
11-9	Reserved	0	Reserved. The reserved bit location is always read as 0. A value written to this field has no effect.
8-4	MINITER	0-31	Minimum number of iterations to be executed
		0	1
		1	1-31
3-2	Reserved	0	Reserved. The reserved bit location is always read as 0. A value written to this field has no effect.
1	EPRORED		Prolog reduction.
		0	Prolog reduction disabled
		1	Prolog reduction enabled
0	EXMASTR		Disable/enable Max Log-MAP.
		0	Max star disabled (enable Max Log-MAP)
		1	Max star enabled (enable log MAP)

6.6 TCP2 Input Configuration Register 4 (TCPIC4)

The TCP2 input configuration register 4 (TCPIC4) is shown in Figure 37 and described in Table 10. TCPIC4 informs the TCP2 on the EDMA3 data flow segmentation.

Figure 37. TCP2 Input Configuration Register 4 (TCPIC4)

31	16
Reserved	
R/W-0	

15	13	12	8	7	6	5	0
Reserved		CRCITERPASS		Reserved		CRCLEN	
R/W-0		R/W-0		R/W-0		R/W-0	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 10. TCP2 Input Configuration Register 4 (TCPIC4) Field Descriptions

Bit	Field	Value	Description
31-13	Reserved	0	Reserved. The reserved bit location is always read as 0. A value written to this field has no effect.
12-8	CRCITERPASS	1-31	Number of consecutive CRC passing iterations required before decoder termination
		0	1
		1	1 to31
7-6	Reserved		Reserved. The reserved bit location is always read as 0. A value written to this field has no effect.
5-0	CRCLEN	0-32	CRC polynomial length
		0	Disable CRC
		1	1 to 32 = CRC polynomial length

6.7 TCP2 Input Configuration Register 5 (TCPIC5)

The TCP2 input configuration register 5 (TCPIC5) is shown in Figure 38 and described in Table 11. TCPIC5 provides the 32-bit CRC Polynomial to TCP2.

Figure 38. TCP2 Input Configuration Register 5 (TCPIC5)

31	0
CRCPOLY	
R/W-0	

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

Table 11. TCP2 Input Configuration Register 5 (TCPIC5) Field Descriptions

Bit	Field	Value	Description
31-0	CRCPOLY	0	CRC polynomial. This 32 bit parameter defines the CRC polynomial. Table 12 lists some examples for CRCPOLY.

Table 12. CRC Examples

CRC Length	CRC Poly	Polynomial
6	00000023	X^6+X^2+X+1
8	000000CD	$X^8+X^7+X^4+X^3+X+1$
10	000003EC	$X^{10}+X^9+X^8+X^7+X^6+X^4+X^3+1$
12	00000F89	$X^{12}+X^{11}+X^{10}+X^9+X^8+X^4+X+1$
16	0000E433	$X^{16}+X^{15}+X^{14}+X^{11}+X^6+X^5+X^2+X+1$
24	00800043	$X^{24}+X^7+X^2+X+1$
32	80000043	$X^{32}+X^7+X^2+X+1$

6.8 Tail Symbols

The tail symbols are symbols appended to the end of the information symbols to force both the non-interleaved encoder to the 0 state and the interleaved encoder to the 0 state. The raw input data streams for the 6-bit tail symbols are shown in the TCP2 input registers TCPIC6 -11.

It also shows how to load the six 18-bit tail registers in standalone mode. These registers are loaded from the EDMA3 bus. In shared mode, input registers TCPIC9 - 11 are not used. Regardless of the mode, these 6 input parameters have to be loaded by the EDMA3 prior to any decoding.

6.9 TCP2 Input Configuration Register 6 (TCPIC6)

The TCP2 input configuration register 6 (TCPIC6) is shown in Figure 39 and described in Table 13. TCPIC6 sets the tail bits used by the TCP.

Figure 39. TCP2 Input Configuration Register 6 (TCPIC6)

31	18	17	0
Reserved		TAIL1	
R/W-0		R/W-0	

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

Table 13. TCP2 Input Configuration Register 6 (TCPIC6) Field Descriptions

Bit	Field	Value	Description
31-18	Reserved	0	Reserved. The reserved bit location is always read as 0. A value written to this field has no effect.
17-0	TAIL1	0-FFFF FFFFh	Tail bit. Values must be set as in the following list.

- CDMA-2000 Tail Symbol Pattern for Code Rate 1/5

tail+2	tail+1	tail+0
(x10+x10+x10)/3 (x10+x10+x10)/3	(x10+x10+x10)/3	

- CDMA-2000 Tail Symbol Pattern for Code Rate 1/4, 1/3

tail+2	tail+1	tail+0
(x10+x10)/2	(x10+x10)/2	(x10+x10)/2

- CDMA-2000 Tail Symbol Pattern for Code Rate 1/2 or 3/4

tail+2	tail+1	tail+0
x10	x10	x10

- CDMA-2000 Tail Symbol Pattern for Code Rate 1/5 or 1/4

tail+2	tail+1	tail+0
x10	x10	x10

- CDMA-2000 Tail Symbol Pattern for Code Rate 1/3, 1/2, or 3/4

tail+2	tail+1	tail+0
x10	x10	x10

6.10 TCP2 Input Configuration Register 7 (TCPIC7)

The TCP2 input configuration register 7 (TCPIC7) is shown in Figure 40 and described in Table 14. TCPIC7 sets set the tail bits used by the TCP.

Figure 40. TCP2 Input Configuration Register 7 (TCPIC7)

31	18	17	0
Reserved		TAIL2	
R/W-0		R/W-0	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 14. TCP2 Input Configuration Register 7 (TCPIC7) Field Descriptions

Bit	Field	Value	Description
31-18	Reserved	0	Reserved. The reserved bit location is always read as 0. A value written to this field has no effect.
17-0	TAIL2	0-FFFF FFFFh	Tail bit. Values must be set as in the following list.

- CDMA-2000 Tail Symbol Pattern for Code Rate 1/4

tail+2	tail+1	tail+0
p10	p10	p10

- CDMA-2000 Tail Symbol Pattern for Code Rate 1/3 or 1/2

tail+2	tail+1	tail+0
p10	p10	p10

- CDMA-2000 Tail Symbol Pattern for Code Rate 3/4

tail+2	tail+1	tail+0
0	0	p10

- CDMA-2000 Tail Symbol Pattern for Code Rate 1/5 or 1/4

tail+2	tail+1	tail+0
p10	p10	p10

- CDMA-2000 Tail Symbol Pattern for Code Rate 1/3, 1/2, or 3/4

tail+2	tail+1	tail+0
p10	p10	p10

6.11 TCP2 Input Configuration Register 8 (TCPIC8)

The TCP2 input configuration register 8 (TCPIC8) is shown in Figure 41 and described in Table 15. TCPIC8 sets the tail bits used by the TCP.

Figure 41. TCP2 Input Configuration Register 8 (TCPIC8)

31	18	17	0
Reserved		TAIL3	
R/W-0		R/W-0	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 15. TCP2 Input Configuration Register 8 (TCPIC8) Field Descriptions

Bit	Field	Value	Description
31-18	Reserved	0	Reserved. The reserved bit location is always read as 0. A value written to this field has no effect.
17-0	TAIL3	0-FFFF FFFFh	Tail bit. Values must be set as in the following list.

- CDMA-2000 Tail Symbol Pattern for Code Rate 1/5

tail+2	tail+1	tail+0
p11	p11	p11

- CDMA-2000 Tail Symbol Pattern for Code Rate 1/4

tail+2	tail+1	tail+0
p11	p11	p11

- CDMA-2000 Tail Symbol Pattern for Code Rate 1/3

tail+2	tail+1	tail+0
0	0	0

- CDMA-2000 Tail Symbol Pattern for Code Rate 1/5 or 1/4

tail+2	tail+1	tail+0
p11	p11	p11

- CDMA-2000 Tail Symbol Pattern for Code Rate 1/3, 1/2, or 3/4

tail+2	tail+1	tail+0

6.12 TCP2 Input Configuration Register 9 (TCPIC9)

The TCP2 input configuration register 9 (TCPIC9) is shown in Figure 42 and described in Table 16. TCPIC9 sets the tail bits used by the TCP.

Figure 42. CP2 Input Configuration Register 9 (TCPIC9)

31	18	17	0
Reserved		TAIL4	
R/W-0		R/W-0	

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

Table 16. CP2 Input Configuration Register 9 (TCPIC9) Field Descriptions

Bit	Field	Value	Description
31-18	Reserved	0	Reserved. The reserved bit location is always read as 0. A value written to this field has no effect.
17-0	TAIL4	0-FFFF FFFFh	Tail bit. Values must be set as in the following list.

- CDMA-2000 Tail Symbol Pattern for Code Rate 1/5

tail+2	tail+1	tail+0
(x20+x20+x20)/3 (x20+x20+x20)/3	(x20+x20+x20)/3	

- CDMA-2000 Tail Symbol Pattern for Code Rate 1/4, 1/3

tail+2	tail+1	tail+0
(x20+x20)/2	(x20+x20)/2	(x20+x20)/2

- CDMA-2000 Tail Symbol Pattern for Code Rate 1/2 or 3/4

tail+2	tail+1	tail+0
x20	x20	x20

- CDMA-2000 Tail Symbol Pattern for Code Rate 1/5 or 1/4

tail+2	tail+1	tail+0
x20	x20	x20

- CDMA-2000 Tail Symbol Pattern for Code Rate 1/3, 1/2, or 3/4

tail+2	tail+1	tail+0
x20	x20	x20

6.13 TCP2 Input Configuration Register 10 (TCPIC10)

The TCP2 input configuration register 10 (TCPIC10) is shown in Figure 43 and described in Table 17. TCPIC10 sets the tail bits used by the TCP.

Figure 43. TCP2 Input Configuration Register 10 (TCPIC10)

31	18	17	0
Reserved		TAIL5	
R/W-0		R/W-0	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 17. TCP2 Input Configuration Register 10 (TCPIC10) Field Descriptions

Bit	Field	Value	Description
31-18	Reserved	0	Reserved. The reserved bit location is always read as 0. A value written to this field has no effect.
17-0	TAIL5	0-FFFF FFFFh	Tail bit. Values must be set as in the following list.

- CDMA-2000 Tail Symbol Pattern for Code Rate 1/5

tail+2	tail+1	tail+0
p20	p20	p20

- CDMA-2000 Tail Symbol Pattern for Code Rate 1/4

tail+2	tail+1	tail+0
p20	p20	p20

- CDMA-2000 Tail Symbol Pattern for Code Rate 1/3

tail+2	tail+1	tail+0
p20	p20	p20

- CDMA-2000 Tail Symbol Pattern for Code Rate 3/4

tail+2	tail+1	tail+0
0	0	p20

- CDMA-2000 Tail Symbol Pattern for Code Rate 1/5 or 1/4

tail+2	tail+1	tail+0
p20	p20	p20

- CDMA-2000 Tail Symbol Pattern for Code Rate 1/3, 1/2 or 3/4

tail+2	tail+1	tail+0
p20	p20	p20

6.14 TCP2 Input Configuration Register 11 (TCPIC11)

The TCP2 input configuration register 11 (TCPIC11) is shown in Figure 44 and described in Table 18. TCPIC11 sets the tail bits used by the TCP.

Figure 44. TCP2 Input Configuration Register 11 (TCPIC11)

31	18	17	0
Reserved		TAIL6	
R/W-0		R/W-0	

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

Table 18. TCP2 Input Configuration Register 11 (TCPIC11) Field Descriptions

Bit	Field	Value	Description
31-18	Reserved	0	Reserved. The reserved bit location is always read as 0. A value written to this field has no effect.
17-0	TAIL6	0-FFFF FFFFh	Tail bit. Values must be set as in the following list.

- CDMA-2000 Tail Symbol Pattern for Code Rate 1/5

tail+2	tail+1	tail+0
p21	p21	p21

- CDMA-2000 Tail Symbol Pattern for Code Rate 1/4

tail+2	tail+1	tail+0
p21	p21	p21

- CDMA-2000 Tail Symbol Pattern for Code Rate 1/3

tail+2	tail+1	tail+0
0	0	0

- CDMA-2000 Tail Symbol Pattern for Code Rate 1/5 or 1/4

tail+2	tail+1	tail+0
p21	p21	p21

- CDMA-2000 Tail Symbol Pattern for Code Rate 1/3, 1/2, or 3/4

tail+2	tail+1	tail+0
0	0	0

6.15 TCP2 Input Configuration Register 12 (TCPIC12)

The TCP2 input configuration register 12 (TCPIC12) is shown in Figure 45 and described in Table 19.

Figure 45. TCP2 Input Configuration Register 12 (TCPIC12)

31	24	23	0
Reserved		EXT_SCALE_0_3	
R/W-0		R/W-0	

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

Table 19. TCP2 Input Configuration Register 12 (TCPIC12) Field Descriptions

Bit	Field	Value	Description
31-24	Reserved	0	Reserved. The reserved bit location is always read as 0. A value written to this field has no effect.
23-0	EXT_SCALE_0_3	0-FFFF FFFFh	Extrinsic scale factor
		23:18	Extrinsic scale factor 3
		17:12	Extrinsic scale factor 2
		11:6	Extrinsic scale factor 1
		5:0	Extrinsic scale factor 0

6.16 TCP2 Input Configuration Register 13 (TCPIC13)

The TCP2 input configuration register 13 (TCPIC13) is shown in Figure 46 and described in Table 20.

Figure 46. TCP2 Input Configuration Register 13 (TCPIC13)

31	24	23	0
Reserved		EXT_SCALE_4_7	
R/W-0		R/W-0	

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

Table 20. TCP2 Input Configuration Register 13 (TCPIC13) Field Descriptions

Bit	Field	Value	Description
31-24	Reserved	0	Reserved. The reserved bit location is always read as 0. A value written to this field has no effect.
23-0	EXT_SCALE_4_7	0-FFFF FFFFh	Extrinsic scale factor
		23:18	Extrinsic scale factor 7
		17:12	Extrinsic scale factor 6
		11:6	Extrinsic scale factor 5
		5:0	Extrinsic scale factor 4

6.17 TCP2 Input Configuration Register 14 (TCPIC14)

The TCP2 input configuration register 14 (TCPIC14) is shown in Figure 47 and described in Table 21.

Figure 47. TCP2 Input Configuration Register 14 (TCPIC14)

31 24	23 0
Reserved	EXT_SCALE_8_11
R/W-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

Table 21. TCP2 Input Configuration Register 14 (TCPIC14) Field Descriptions

Bit	Field	Value	Description
31-24	Reserved	0	Reserved. The reserved bit location is always read as 0. A value written to this field has no effect.
23-0	EXT_SCALE_8_11	0-FFFF FFFFh	Extrinsic scale factor
		23:18	Extrinsic scale factor 11
		17:12	Extrinsic scale factor 10
		11:6	Extrinsic scale factor 9
		5:0	Extrinsic scale factor 8

6.18 TCP2 Input Configuration Register 15 (TCPIC15)

The TCP2 input configuration register 15 (TCPIC15) is shown in Figure 48 and described in Table 22.

Figure 48. TCP2 Input Configuration Register 15 (TCPIC15)

31 24	23 0
Reserved	EXT_SCALE_12_15
R/W-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

Table 22. TCP2 Input Configuration Register 15 (TCPIC15) Field Descriptions

Bit	Field	Value	Description
31-24	Reserved	0	Reserved. The reserved bit location is always read as 0. A value written to this field has no effect.
23-0	EXT_SCALE_12_15	0-FFFF FFFFh	Extrinsic scale factor
		23:18	Extrinsic scale factor 15
		17:12	Extrinsic scale factor 14
		11:6	Extrinsic scale factor 13
		5:0	Extrinsic scale factor 12

The 16 extrinsic scale registers are 6 bits each and have a (1,5) fixed-point precision. The unsigned fixed-point numbers can range from 0.0 to 1.0. For example, 0.5 is equal to 0.10000 or 0x10. These registers are only used if e_max_star is 0. If e_max_star is 1, then the extrinsic scale factor is automatically set to a 1.0. The reset value for each register is 1.0 or 0x20.

The 16 extrinsic scale registers are selected depending on the iteration number and active MAP as shown in Table 23. MAP 0 is the non-interleaved MAP decode and MAP1 is the interleaved MAP decode.

Table 23. Extrinsic Scale Registers

Iteration Number	MAP	Extrinsic Scaling Register
0	0	0
0	1	1
1	0	2
1	1	3
2	0	4
2	1	5
3	0	6
3	1	7
4	0	8
4	1	9
5	0	10
5	1	11
6	0	12
6	1	13
7	0	14
7	1	15
8	0	15
8	1	15
.	.	.
.	.	.
.	.	.
31	1	15

6.19 TCP2 Output Parameter Register 0 (TCPOUT0)

The TCP2 output parameter register 0 (TCPOUT0) is shown in Figure 49 and described in Table 24.

Figure 49. TCP2 Output Parameter Register 0 (TCPOUT0)

31 29	28 24	23 20	19 0
Reserved	FINAL_ITER	Reserved	SNR_M1
R/W-0	R/W-0	R/W-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 24. TCP2 Output Parameter Register 0 (TCPOUT0) Field Descriptions

Bit	Field	Value	Description
31-29	Reserved	0	Reserved. The reserved bit location is always read as 0. A value written to this field has no effect.
28-24	FINAL_ITER	0-FFFFh	Number of decoded iterations.
23-20	Reserved	0	Reserved. The reserved bit location is always read as 0. A value written to this field has no effect.
19-0	SNR_M1	0	First moment of SNR calculations.

6.20 TCP2 Output Parameter Register 1 (TCPOUT1)

The TCP2 output parameter register 1 (TCPOUT1) is shown in Figure 50 and described in Table 25.

Figure 50. TCP2 Output Parameter Register 1 (TCPOUT1)

31 30	28	28	27 24
SNR_EXCEED	CRC_PASS	ACTIVE_MAP	Reserved
R/W-0	R/W-0	R/W-0	R/W-0

23 0
SNR_M2
R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 25. TCP2 Output Parameter Register 1 (TCPOUT1) Field Descriptions

Bit	Field	Value	Description
31-30	SNR_EXCEED		Decoder terminated due to crc
		0	0 MAP0 failed SNR
		0	1 MAP0 passed SNR
		1	0 MAP0 failed SNR
		1	1 MAP0 passed SNR
29	CRC_PASS		Decoder terminated due to snr
		0	Crc has not passed
		1	Crc has passed
28	ACTIVE_MAP		Active map
		0	MAP0 is active
		1	MAP1 is active
			Note: ACTIVE_MAP bit status is reserved when the FREE bit = 0 and the SOFT bit = 0.
27-24	Reserved		Reserved. The reserved bit location is always read as 0. A value written to this field has no effect.
23-0	SNR_M2		Second moment of calculation

6.21 TCP2 Output Parameter Register 2 (TCPOUT2)

The TCP2 output parameter register 2 (TCPOUT2) is shown in Figure 51 and described in Table 26.

Figure 51. TCP2 Output Parameter Register 2 (TCPOUT2)

31	16	15	0
CNT_RE_MAP1		CNT_RE_MAP0	
R/W-0		R/W-0	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 26. TCP2 Output Parameter Register 2 (TCPOUT2) Field Descriptions

Bit	Field	Value	Description
31-16	CNT_RE_MAP1		Number of MAP1 re-encode errors
15-0	CNT_RE_MAP0		Number of MAP0 re-encode errors

6.22 TCP2 Execution Register (TCPEXE)

The TCP2 execution register (TCPEXE) is shown in Figure 52 and described in Table 27.

Figure 52. TCP2 Execution Register (TCPEXE)

31	16
Reserved	
W-0	

15	3	2	0
Reserved		EXECUTION_INSTR	
W-0		W-0	

LEGEND: R/W = Read/Write; R = Read only; W = Write only; -*n* = value after reset

Table 27. TCP2 Execution Register (TCPEXE) Field Descriptions

Bit	Field	Value	Description
31-3	Reserved	0	Reserved. The reserved bit location is always read as 0. A value written to this field has no effect.
2-0	EXECUTION_INSTR		Execution commands of TCP.
		0	No instruction
		1	Start TCP
		4	Debug mode. Normal initialization and wait in MAP state 0.
		5	Debug mode. Execute one MAP decode and wait in MAP state 6.
		6	Debug mode. Execute remaining MAP decodes and complete normal ending.
		7	SOFT RESET. Soft reset all TCP registers except for the execution register and endian register.

6.23 TCP2 Endian Register (TCPEND)

The TCP2 endian register (TCPEND) is shown in Figure 53 and described in Table 28. TCPEND should only be used when the DSP is set to big-endian mode.

Figure 53. TCP2 Endian Register (TCPEND)

31	8
Reserved	
R/W-0	

7	2	1	0
Reserved		ENDIAN_ EXTR	ENDIAN_ INTR
R/W-0		R/W-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 28. TCP2 Endian Register (TCPEND) Field Descriptions

Bit	Field	Value	Description
31-2	Reserved		Reserved. The reserved bit location is always read as 0. A value written to this field has no effect.
1	ENDIAN_EXTR		Endianness view of extrinsic table.
		0	3,2,1,0,7,6,5,4 ⇒ 7,6,5,4,3,2,1,0 (bytes)
		1	0,1,2,3,4,5,6,7 ⇒ 7,6,5,4,3,2,1,0 (bytes)
0	ENDIAN_INTR		Endianness view of interleaver table. No effect if in little endian mode.
		0	1,0,3,2 ⇒ 3,2,1,0 (halfwords)
		1	0,1,2,3 ⇒ 3,2,1,0 (halfwords)

6.24 TCP2 Error Register (TCPERR)

The TCP2 error register (TCPERR) is shown in Figure 54 and described in Table 29. In case of an error, the coprocessor sends an interrupt to the TCI648x CPU. The following errors are feedback in the error word.

Figure 54. TCP2 Error Register (TCPERR)

31	16
Reserved	
R-0	

15	14	13	12	11	10	9	8
Reserved	MAX MINTER	Reserved		ACC	OP	INT	SNR
R-0	R-0	R-0		R-0	R-0	R-0	R-0

7	6	5	4	3	2	1	0
R	Reserved		SF	Reserved	P	F	ERR
R-0	R-0		R-0	R-0	R-0	R-0	R-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 29. TCP2 Error Register (TCPERR) Field Descriptions

Bit	Field	Value	Description
31-15	Reserved		Reserved. The reserved bit location is always read as 0. A value written to this field has no effect.
14	MAXMINTER		Max/min iteration
		0	No error
		1	Min_iter > max_iter
13-12	Reserved		Reserved. The reserved bit location is always read as 0. A value written to this field has no effect.
11	ACC		Spurious TCP memory access.
		0	No error.
		1	TCP memory access error: System & Parities, Hard Decision, Extrinsic, or Apriori memories were accessed at the wrong time. Chapter 9 describes the process by which the system determines when EDMA3 can access these memories. The memories should not be viewed except when TCPEXE is set for Debug Mode and the DSP has halted.
10	OP		Output parameters load for shared processing.
		0	No error
		1	Output parameter read flag is set in SP mode
9	INT		Interleaver table load for shared processing.
		0	No error
		1	There has been a request to load the interleaver table (interleaver flag bit set to 1)
8	SNR		SNR threshold
		0	No error
		1	SNR threshold > 100
7	R		Reliability length
		0	No error
		1	Reliability length error due to one of the following: • Reliability length < 40 • sw_r != 128 in SP mode if long_type = 1 or 2 • sw_r < 65 in SP mode if long_type = 3
6-5	Reserved	0	Reserved. The reserved bit location is always read as 0. A value written to this field has no effect.

Table 29. TCP2 Error Register (TCPERR) Field Descriptions (continued)

Bit	Field	Value	Description
4	SF		Subframe length.
		0	No error
		1	Subframe length > 5114
3	Reserved	0	Reserved. The reserved bit location is always read as 0. A value written to this field has no effect.
2	P		Prolog length.
		0	No error
		1	Prolog length < 4 or> 48
1	F		Frame length.
		0	No error
		1	Frame length error, due to one of the following: • Standalone mode: frame length < 40, or frame length > 20730 • Shared processing mode: indicates that frame length < 256 or frame length > 20480 and f% 256 = 0 if long_type = 1 or 2 • f < 128 or f > 20480 in SP mode if long_type = 3
0	ERR		Error status.
		0	No error
		1	Error detected in the input registers

6.25 TCP2 Status Register (TCPSTAT)

The TCP2 status register (TCPSTAT) is shown in Figure 55 and described in Table 30.

Figure 55. TCP2 Status Register (TCPSTAT)

31	28	27	24
Reserved		TCP_STATE	
R-0		R-0	

23	22	21	20	16
CRC_PASS	SNR_EXCEED		ACTIVE_ITER	
R-0	R-0		R-0	

15	12	11	10	9	8
ACTIVE_STATE		ACTIVE_ MAP	EMUHALT	ROP	RHD
R-0		R-0	R-0	R-0	R-0

7	6	5	4	3	2	1	0
REXT	WAP	WSP	WINT	WIC	ERR	DEC_BUSY	Reserved
R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

Table 30. TCP2 Status Register (TCPSTAT) Field Descriptions

Bit	Field	Value	Description
31-28	Reserved	0	Reserved. The reserved bit location is always read as 0. A value written to this field has no effect.
27-24	TCP_STATE		TCP2 top level state of state machine. The states are defined in the TCP2 state machine section.
23	CRC_PASS		CRC status
		0	CRC has not passed
		1	CRC passed
22-21	SNR_EXCEED		SNR status
		0	0 MAP0 failed SNR
		0	1 MAP0 passed SNR
		1	0 MAP1 failed SNR
		1	1 MAP1 passed SNR
20-16	ACTIVE_ITER		Active TCP2 iteration status.
15-12	ACTIVE_STATE		Active state status
11	ACTIVE_MAP		Active map status
			Note: ACTIVE_MAP bit status is reserved when the FREE bit = 0 and the SOFT bit = 0.
10	EMUHALT		Defines if the TCP2 is halted due to emulation.
		0	Not halted due to emulation
		1	Halted due to emulation
			Note: EMUHALT bit status is reserved when the FREE bit = 0 and the SOFT bit = 1.
9	ROP		Defines if the TCP2 is waiting for output parameter data to be read.
		0	Not waiting
		1	Waiting for RAM output registers to be read
8	RHD		Defines if the TCP2 is waiting for hard decision data to be read.
		0	Not waiting
		1	Waiting for RAM output/decision memory to be read

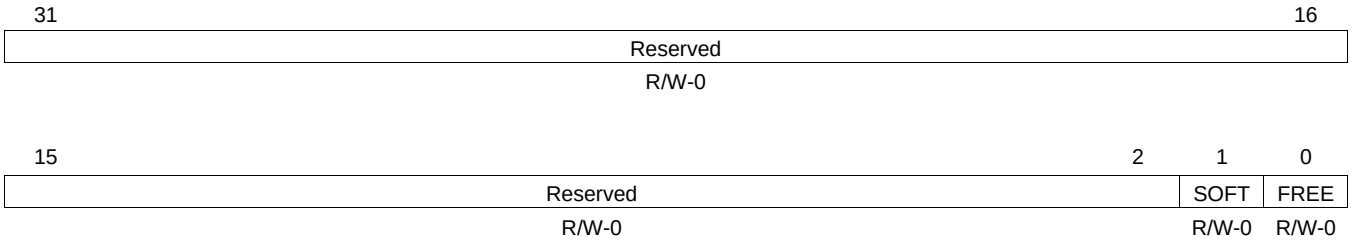
Table 30. TCP2 Status Register (TCPSTAT) Field Descriptions (continued)

Bit	Field	Value	Description
7	REXT		Defines if the TCP2 is waiting for extrinsic memory 0 data to be read.
		0	Not waiting
		1	Waiting for RAM extrinsic memory 0 to be read
6	WAP		Defines if the TCP2 is waiting for a extrinsic memory 1 data to be written.
		0	Not waiting
		1	Waiting for RAM extrinsic memory 1 to be loaded
5	WSP		Defines if the TCP2 is waiting for systematic and parity data to be written.
		0	Not waiting
		1	Waiting for RAM data/system and parity memory to be loaded
4	WINT		Defines if the TCP2 is waiting for interleaver indexes to be written.
		0	Not waiting
		1	Waiting for RAM interleaver memory to be loaded
3	WIC		Defines if the TCP2 is waiting for input control words to be written.
		0	Not waiting
		1	Waiting for input register to be loaded
2	ERR		Defines if the TCP2 has encountered an error.
		0	No input register error
		1	Input register error
1	DEC_BUSY		Decoder status
		0	MAP decoder is in state 0
		1	MAP decoder is in state 1 to 8
0	Reserved		Reserved. The reserved bit location is always read as 0. A value written to this field has no effect.

6.26 TCP2 Emulation Register (TCPEMU)

In emulation mode, the access to TCP2 memories can be done in read or write. TCP2 supports emulation mode. Emulation support helps in system debug. Emulation modes are achieved with the programmable SOFT and FREE bits in the TCP2 Emulation Register (TCPEMU) at the configuration bus address 0x00070. The TCP2 emulation register (TCPEMU) is shown in Figure 56 and described in Table 31.

Figure 56. TCP2 Emulation Register (TCPEMU)



LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 31. TCP2 Emulation Register (TCPEMU) Field Descriptions

Bit	Field	Value	Description
31-2	Reserved	0	Reserved. The reserved bit location is always read as 0. A value written to this field has no effect.
1	SOFT		SOFT bit
		0	Emulation halt at either end of MAP decode or at end of decode prior to last XEVT. Stop at the end of MAP decode has priority.
		1	Emulation halt at end of decode prior to last XEVT
0	FREE		FREE bit
		0	Soft emulation bit takes effect
		1	TCP2 ignores emulation halt and runs to completion

The FREE and SOFT bits are designed to enable a flexible method of how the TCP2 is operated during an emulation halt of the CPU. The FREE bit determines if an emulation halt of the CPU will halt the TCP2 at all. If the FREE bit is set, and an emulation halt of the CPU occurs, the TCP2 will continue processing normally. If the FREE bit is cleared, and an emulation halt of the CPU occurs, then TCP2 will be halted in a manner determined by the SOFT bits setting. Note that when FREE = 1, SOFT has no effect.

Given that FREE = 0, and an emulation halt of the CPU occurs, the TCP2 will halt as follows based on the setting of SOFT bit.

SOFT = 0:

Emulation halt uses TCP2 debug mode. Any current MAP processing must complete before entering the emulation mode.

Current data transfer on the bus should complete and pending read/write requests to/from CPU/DMA should complete before emulation halt. If an active output event(TCPREVT/TCPXEVT) is output before this emulation halt, it should service that request before going into a suspend state. If an active MAP is processing before this emulation halt, TCP2 should service that request before going into a suspend state. TCP2 will pause after each MAP processing. No new read/write events to CPU or DMA should be generated. Any ongoing CPU or DMA read/write services to TCP2 should complete. The TCP2 will restart from the halt state and it will run to normal completion until next emulation halt.

SOFT = 1:

Current data transfer on the bus should complete and pending read/write requests to/from CPU/DMA should complete before emulation halt. If an active frame is processing before this emulation halt, it should service all requests before going into suspend state. Any ongoing CPU or DMA read/write services to TCP2 should complete. Frame processing should complete.

The TCP2 is halted (or paused) after processing the ongoing frame. Any current frame processing must complete. Sync vents for the new frame will be hold until TCP_EMUSUSP is released. The TCP2 is restarted from the paused state and begins the next frame operations.

In TCP_STATE = 14, the TCP_EMUSUSP will have no effect. The TCP2 will go to the next state (TCP_STATE=0) and then the emususp will be processed.

In TCP_STATE = 0, the TCP_EMUSUSP will cause the emuack to go active if no EDMA3 transactions are active.

In TCP_STATE = 1, the TCP_EMUSUSP will cause the emuack to go active if no EDMA3 transactions are active. The memory_access error bit will not go active if emuack = 1, and the tcp_int will not trigger if the memories are accessed while emuack = 1.

The emususp_rt signal is not used in the TCP2. Bit[2](RT_SEL) for the emulation register is not included and the bit is reserved.

7 Endianness

The endianness manager is responsible for managing the endianness of data when DSP is configured in big endian mode. When the DSP is configured in little-endian mode, the endianness manager has no effect.

This architecture supports both big- and little-endian operation.

The TCP2 always works in little-endian mode, the input/output data to/from the processing unit is always in little-endian format. Therefore, the role of the endianness manager is to order the data correctly when the DSP is configured in big-endian mode.

For the data represented on the configuration (CFG) data bus, byte endianness is not an issue. The endianness manager has no effect on 32-bit data on the CFG bus.

In all cases except for interleaver indexes and extrinsics, the endianness manager swaps the words within the double-word for all TCP2 incoming 64-bit data (Figure 57 and Figure 58) and all TCP2 outgoing 64-bit data (Figure 59 and Figure 60).

Figure 57. Data Source - EDMA3 (Big Endian)

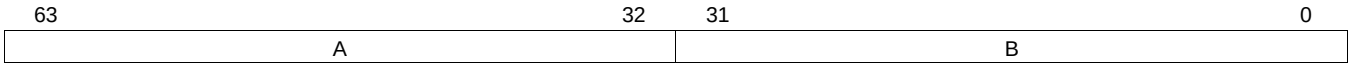


Figure 58. Data Destination - Kernel (Little Endian)

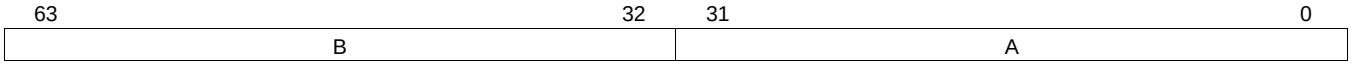


Figure 59. Data Source - Kernel (Little Endian)

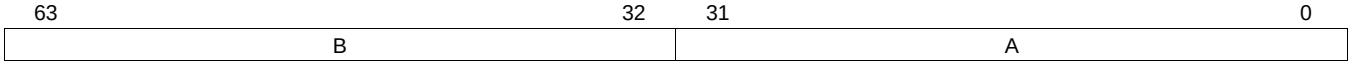
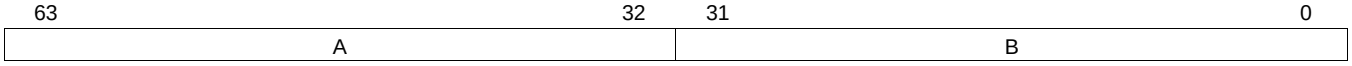


Figure 60. Data Destination - EDMA3 (Big Endian)



7.1 Data Memory for Systematic

Figure 61. Data Memory

63:62	61:56	55:50	49:44	43:38	37:32	31:30	29:24	23:18	17:12	11:6	5:0
RSVD	SP9	SP8	SP7	SP6	SP5	RSVD	SP4	SP3	SP2	SP1	SP0

Figure 62. EN = 1 (Little-Endian Mode) Rate = 1/2

Word N + 1					Word N				
SP9 0	SP8 A1'	SP7 0	SP6 0	SP5 X1	SP4 0	SP3 0	SP3 0	SP3 0	SP0 X0
Word N + 3					Word N + 2				
SP9 0	SP8 A3'	SP7 0	SP6 0	SP5 X3	SP4 0	SP3 0	SP3 0	SP3 A2	SP0 X2

Figure 63. EN = 0 (Big-Endian Mode) Rate = 1/2

Word N					Word N + 1				
SP4 0	SP3 0	SP2 0	SP1 A0	SP0 X0	SP9 0	SP8 A1'	SP7 0	SP6 0	SP5 X1
Word N + 2					Word N + 3				
SP4 0	SP3 0	SP2 0	SP1 A2	SP0 X2	SP9 0	SP8 A3'	SP7 0	SP6 0	SP5 X3

Figure 64. EN = 1 (Little-Endian Mode) Rate = 1/3

Word N + 1					Word N				
SP9 0	SP8 A1'	SP7 0	SP6 A1	SP5 X1	SP4 0	SP3 A0'	SP2 0	SP1 A0	SP0 X0
Word N + 3					Word N + 2				
SP9 0	SP8 A3'	SP7 0	SP6 A3	SP5 X3	SP4 0	SP3 A2'	SP2 0	SP1 A2	SP0 X2

Figure 65. EN = 0 (Big-Endian Mode) Rate = 1/3

Word N					Word N + 1				
SP4 0	SP3 A0'	SP2 0	SP1 A0	SP0 X0	SP9 0	SP8 A1'	SP7 0	SP6 A1	SP5 X1
Word N + 2					Word N + 3				
SP4 0	SP3 A2'	SP2 0	SP1 A2	SP0 X2	SP9 0	SP8 A3'	SP7 0	SP6 A3	SP5 X3

Figure 66. EN = 1 (Little-Endian Mode) Rate = 1/4

Word N + 1					Word N				
SP9 B1'	SP8 A1'	SP7 0	SP6 A1	SP5 X1	SP4 B0'	SP3 0	SP2 B0	SP1 A0	SP0 X0
Word N + 3					Word N + 2				
SP9 B3'	SP8 A3'	SP7 0	SP6 A3	SP5 X3	SP4 B2'	SP3 0	SP2 B2	SP1 A2	SP0 X2

Figure 67. EN = 0 (Big-Endian Mode) Rate = 1/4

Word N					Word N + 1				
SP4 B0'	SP3 0	SP2 B0	SP1 A0	SP0 X0	SP9 B1'	SP8 A1'	SP7 0	SP6 A1	SP5 X1
Word N + 2					Word N + 3				
SP4 B2'	SP3 0	SP2 B2	SP1 A2	SP0 X2	SP9 B3'	SP8 A3'	SP7 0	SP6 A3	SP5 X3

Figure 68. EN = 1 (Little-Endian Mode) Rate = 1/5

Word N + 1					Word N				
SP9 B1'	SP8 A1'	SP7 B1	SP6 A1	SP5 X1	SP4 B0'	SP3 A0'	SP2 B0	SP1 A0	SP0 X0
Word N + 3					Word N + 2				
SP9 B3'	SP8 A3'	SP7 B3'	SP6 A3	SP5 X3	SP4 B2'	SP3 A2'	SP2 B2	SP1 A2	SP0 X2

Figure 69. EN = 0 (Big-Endian Mode) Rate = 1/5

Word N					Word N + 1				
SP4 B0'	SP3 A0'	SP2 B0	SP1 A0	SP0 X0	SP9 B1'	SP8 A1'	SP7 B1	SP6 A1	SP5 X1
Word N + 2					Word N + 3				
SP4 B2'	SP3 A2	SP2 B2	SP1 A2	SP0 X2	SP9 B3'	SP8 A3'	SP7 B3	SP6 A3	SP5 X3

Figure 70. EN = 1 (Little-Endian Mode) Rate = 3/4

Word N + 1					Word N				
SP9 0	SP8 0	SP7 0	SP6 0	SP5 X1	SP4 0	SP3 0	SP2 0	SP1 A0	SP0 X0
Word N + 3					Word N + 2				
SP9 0	SP8 A3'	SP7 0	SP6 0	SP5 X3	SP4 0	SP3 0	SP2 0	SP1 0	SP0 X2
Word N + 5					Word N + 4				
SP9 0	SP8 0	SP7 0	SP6 0	SP5 X3	SP4 0	SP3 0	SP2 0	SP1 0	SP0 X2

Figure 71. EN = 0 (Big-Endian Mode) Rate = 3/4

Word N					Word N + 1				
SP4 0	SP3 0	SP2 0	SP1 A0	SP0 X0	SP9 0	SP8 0	SP7 0	SP6 0	SP5 X1
Word N + 2					Word N + 3				
SP4 0	SP3 0	SP2 0	SP1 0	SP0 X2	SP9 0	SP8 A3'	SP7 0	SP6 0	SP5 X3
Word N + 4					Word N + 5				
SP4 0	SP3 0	SP2 0	SP1 0	SP0 X4	SP9 0	SP8 0	SP7 0	SP6 0	SP5 X5

7.1.1 Hard Decision Data

1. OUT_ORDER = 0 EN = 1 (Little-Endian Mode)
   OUT_ORDER = 0 results in ordering the hard decision data from 0 to 31 in the 32-bit word output.

Figure 72. Source of Endianness Manager - Ordering of Hard Decisions in 32-Bit Word (OUT_ORDER = 0)

63	62		32	31		1	0
Stage N	Stage N - 1		Stage N - 31	Stage N - 32		Stage N - 62	Stage N - 63

Figure 73. Destination of Endianness Manager - Ordering of Hard Decisions in 32-Bit Word (OUT_ORDER = 0)

63	62		32	31		1	0
Stage N	Stage N - 1		Stage N - 31	Stage N - 32		Stage N - 62	Stage N - 63

2. OUT_ORDER = 1 EN = 1 (Little-Endian Mode)
   OUT_ORDER = 1 results in ordering the hard decision data from 32 to 0 in the 32-bit word output.

Figure 74. Source of Endianness Manager - Ordering of Hard Decisions in 32-Bit Word (OUT_ORDER = 1)

63	62		32	31		1	0
Stage N	Stage N - 1		Stage N - 31	Stage N - 32		Stage N - 62	Stage N - 63

Figure 75. Destination of Endianness Manager - Ordering of Hard Decisions in 32-Bit Word (OUT_ORDER = 1)

63	62		32	31		1	0
Stage N - 31	Stage N - 30		Stage N	Stage N - 63		Stage N - 33	Stage N - 32

3. OUT_ORDER = 0, EN = 0 (Big-Endian Mode)

Figure 76. Source of Endianness Manager - Trellis Stage Ordering of Hard Decisions in 32-Bit Word (OUT_ORDER = 0)

63	62		32	31		1	0
Stage N	Stage N - 1		Stage N - 31	Stage N - 32		Stage N - 62	Stage N - 63

Figure 77. Destination of Endianness Manager (OUT_ORDER = 0)

63	62		32	31		1	0
Stage N - 32	Stage N - 33		Stage N - 63	Stage N		Stage N - 30	Stage N - 31

4. OUT_ORDER = 1 EN = 1 (Little-Endian Mode)

Figure 78. Trellis Stage Ordering of Hard Decisions in 32-Bit Word (OUT_ORDER = 1)

63	62		32	31		1	0
Stage N	Stage N - 1		Stage N - 31	Stage N - 32		Stage N - 62	Stage N - 63

Figure 79. Trellis Stage Ordering of Hard Decisions in 32-Bit Word (OUT_ORDER = 1)

63	62		32	31		1	0
Stage N - 63	Stage N - 62		Stage N - 32	Stage N - 31		Stage N - 1	Stage N

Hard decisions are packed in a 32 bit word inside the TCP2. Data will be saved in word format (32 bits) in the DSP.

Table 32. Hard Decisions in DSP Memory

Address (hex bytes)	Data
Base	HD0 (32 hard decisions)
Base + 4	HD1 (32 hard decisions)

They have to be swapped as described in Figure 80 and Figure 81.

Figure 80. Data Source = Kernel

15	8	7	0
HD1		HD0	
R/W		R/W	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Figure 81. Data Destination = EDMA3 EN = 0 (Big-Endian Mode)

15	8	7	0
HD0		HD1	
R/W		R/W	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

7.1.2 TCP_ENDIAN Register for Endianness Manager

TCP2 input/output data are of different widths and storage in the DSP memory subsystem will be different whether they are saved in native or word format. However, EDMA3 will always read and write words.

There is a need to define a way of handling the data depending on whether they are saved in native or words format. Table 33 summarizes the different data formats, as well as the memory-mapped register TCP_ENDIAN programming (see Figure 82).

Figure 82. TCP_ENDIAN Register

31	16
Reserved	
R/W	

15	2	1	0
Reserved		ENDIAN_ EXTR	ENDIAN_ INTR
R/W		R/W	R/W

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 33. TCP_ENDIAN Programming Register

Data	Native Format	DSP Memory Format	TCP_ENDIAN
Interleaver Indexes	16 bits (15 bits right justified)	16 bits NATIVE	ENDIAN_INTR = 1
		Packed on 32 bits	ENDIAN_INTR = 0
Extrinsic Data	8 bits (7 bits right justified	8 bits NATIVE	ENDIAN_EXTR = 1
		Packed on 32 bits	ENDIAN_EXTR = 0

7.1.3 Interleaver Data

Table 34. Interleaver Data

Little_big_endian	ENDIAN_INTR	Description (MSB to LSB)
0	0	1,0,3,2 ⇒ 3,2,1,0 (half words)
0	1	0,1,2,3 ⇒ 3,2,1,0 (half words)
1	0	Endianness manager has no effect 3,2,1,0 ⇒ 3,2,1,0 (half words)
1	1	Endianness manager has no effect 3,2,1,0 ⇒ 3,2,1,0 (half words)

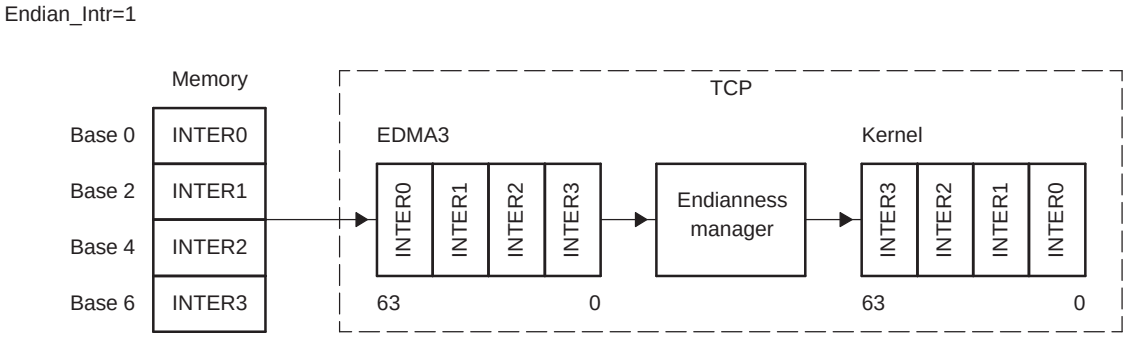
7.1.3.1 ENDIAN_INTR = 1

If ENDIAN_INTR = 1, data are saved in their native format (16 bits) in the DSP (see Table 35).

Table 35. Interleaver Indexes in DSP Memory (ENDIAN_INTR = 1)

Address (hex bytes)	Data
Base	INTER0
Base + 2	INTER1
Base + 4	INTER2
Base + 6	INTER3

Figure 83. Interleaver Indexes in DSP Memory (ENDIAN_INTR = 1)



They have to be swapped as described in Figure 84 and Figure 85.

Figure 84. Data Source - EDMA3 (ENDIAN_INTR = 1)

63 48	47 32	31 16	15 1
INTER0	INTER1	INTER2	INTER3

Figure 85. Data Destination - Kernel (ENDIAN_INTR = 1)

63 48	47 32	31 16	15 1
INTER3	INTER2	INTER1	INTER0

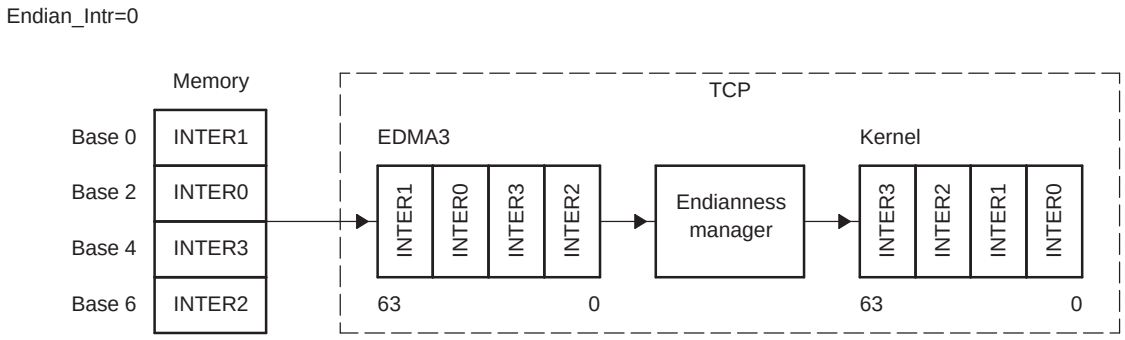
7.1.3.2 ENDIAN_INTR = 0

If ENDIAN_INTR = 0, data are saved in word packed format (32 bits) in the DSP (see Table 36).

Table 36. Interleaver Indexes in DSP Memory (ENDIAN_INTR = 0)

Address (hex bytes)	Data
Base	INTER1
Base + 2	INTER0
Base + 4	INTER3
Base + 6	INTER2

Figure 86. Interleaver Indexes in DSP Memory (ENDIAN_INTR = 0)



They have to be swapped as described in Figure 87 and Figure 88.

Figure 87. Data Source - EDMA3 (ENDIAN_INTR = 0)

63 48	47 32	31 16	15 1
INTER1	INTER0	INTER3	INTER2

Figure 88. Data Destination - Kernel (ENDIAN_INTR = 0)

63 48	47 32	31 16	15 1
INTER3	INTER2	INTER1	INTER0

7.1.4 Extrinsic Data

Table 37. Extrinsic Data

Little_big_endian	ENDIAN_INTR	Description (MSB to LSB)
0	0	3,2,1,0,7,6,5,4 ⇒ 7,6,5,4,3,2,1,0 (bytes)
0	1	0,1,2,3,4,5,6,7 ⇒ 7,6,5,4,3,2,1,0 (bytes)
1	0	Endianness manager has no effect 7,6,5,4,3,2,1,0 ⇒ 7,6,5,4,3,2,1,0 (bytes)
1	1	Endianness manager has no effect 7,6,5,4,3,2,1,0 ⇒ 7,6,5,4,3,2,1,0 (bytes)

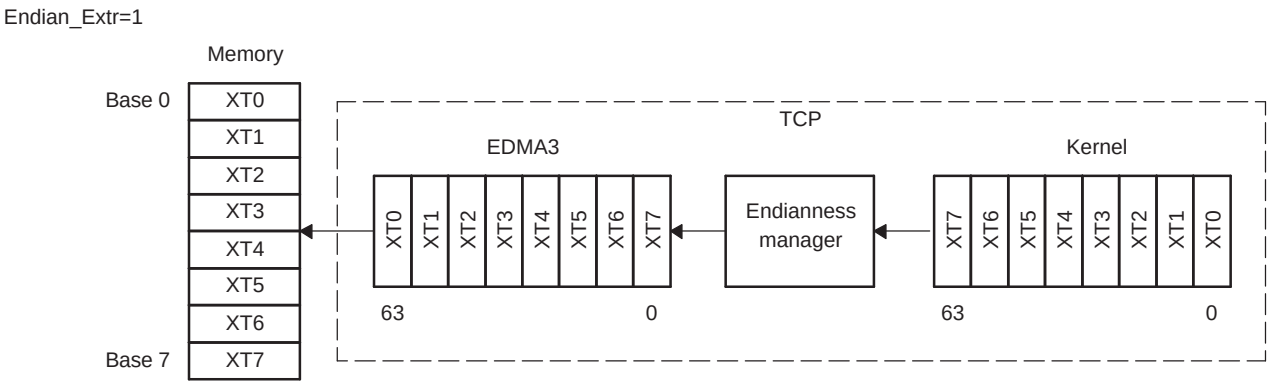
7.1.4.1 ENDIAN_EXTR = 1

If ENDIAN_EXTR = 1, data are saved in their native format (8 bits) in the DSP (see Table 38).

Table 38. Extrinsic in DSP Memory (ENDIAN_EXTR = 1)

Address (hex bytes)	Data
Base	EXT0
Base + 1	EXT1
Base + 2	EXT2
Base + 3	EXT3
Base + 4	EXT4
Base + 5	EXT5
Base + 6	EXT6
Base + 7	EXT7

Figure 89. Extrinsic in DSP Memory (ENDIAN_EXTR = 1)



They have to be swapped as described in Figure 90 and Figure 91.

Figure 90. Data Source - Kernel (ENDIAN_EXTR = 1)

63:56	55:48	47:40	39:32	31:24	23:16	15:8	7:0
EXT7	EXT6	EXT5	EXT4	EXT3	EXT2	EXT1	EXT0

Figure 91. Data Destination - EDMA3 (ENDIAN_EXTR = 1)

63:56	55:48	47:40	39:32	31:24	23:16	15:8	7:0
EXT0	EXT1	EXT2	EXT3	EXT4	EXT5	EXT6	EXT7

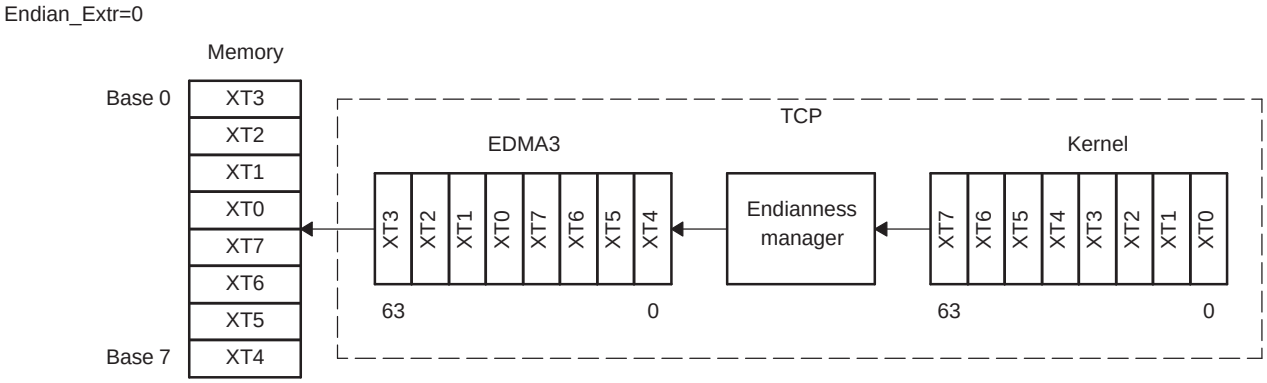
7.1.4.2 ENDIAN_EXTR = 0

If ENDIAN_EXTR = 0, data are saved in word format (32 bits) in the DSP (see Table 39).

Table 39. Extrinsic in DSP Memory (ENDIAN_EXTR = 0)

Address (hex bytes)	Data
Base	EXT3
Base + 1	EXT2
Base + 2	EXT1
Base + 3	EXT0
Base + 4	EXT7
Base + 5	EXT6
Base + 6	EXT5
Base + 7	EXT4

Figure 92. Extrinsic in DSP Memory (ENDIAN_EXTR = 0)



They have to be swapped as described in Figure 93 and Figure 94.

Figure 93. Data Source - Kernel (ENDIAN_EXTR = 0)

63:56	55:48	47:40	39:32	31:24	23:16	15:8	7:0
EXT7	EXT6	EXT5	EXT4	EXT3	EXT2	EXT1	EXT0

Figure 94. Data Destination - EDMA3 (ENDIAN_EXTR = 0)

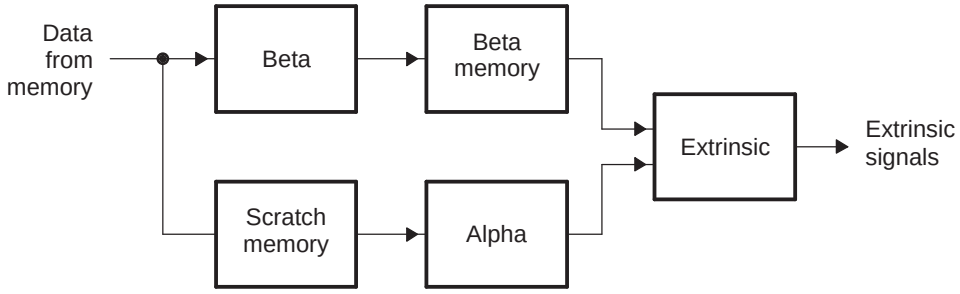
63:56	55:48	47:40	39:32	31:24	23:16	15:8	7:0
EXT3	EXT2	EXT1	EXT0	EXT7	EXT6	EXT5	EXT4

8 Architecture

The TCP2 processing unit (MAP unit) is shown in Figure 95. The beta kernel performs the MAP algorithm backward recursion and produces beta probabilities stored in the beta RAM. The alpha kernel performs the forward recursion producing alpha probabilities, and the extrinsic block computes the extrinsics to be stored in the extrinsics memory.

The processing unit implements the Max* Log-MAP algorithm using a sliding window technique that allows parallel processing and reduces dramatic memory requirements.

Figure 95. MAP Unit Block Diagram



The TCP2 can enable or disable the max star function by modifying the E_MAX_STAR bit in the TCPIC3 register.

- E_MAX_STAR = 0 = Enable max star
- E_MAX_STAR = 1 = Disable max star

Log-map algorithm is implemented in a highly paralleled manner using the sliding window principle. The max-log-map algorithm is implemented with apriori scaled prior to the map decoder.

8.1 Sub-block and Sliding Window Segmentation

The MAP unit splits the frame (in standalone mode) or the sub-frame (in shared processing mode) into sub-blocks.

If the size of the frame is n and the number of states in the encoder trellis is S, then beta states must be stored in the beta memory. To reduce the amount of memory required to store the beta states, the sliding window technique is used. This technique divides the frame size n into several smaller pieces of size r (reliability section). Because the starting state is unknown, a prolog section is added. The initial state of the starting location of the prolog section is unknown. After processing the prolog states and the constraint length, the values of the last prolog states have a high probability of being in the same states as if the states were known from the start. Each sub-block consists of 1or 2 sliding windows. Each sliding window consists of three portions:

- Alpha prolog portion
- Reliability portion
- Beta Prolog portion

Figure 96. Sliding Windows and Sub-blocks Segmentation (Example with 5 Sub-blocks, frame length ≤20730)

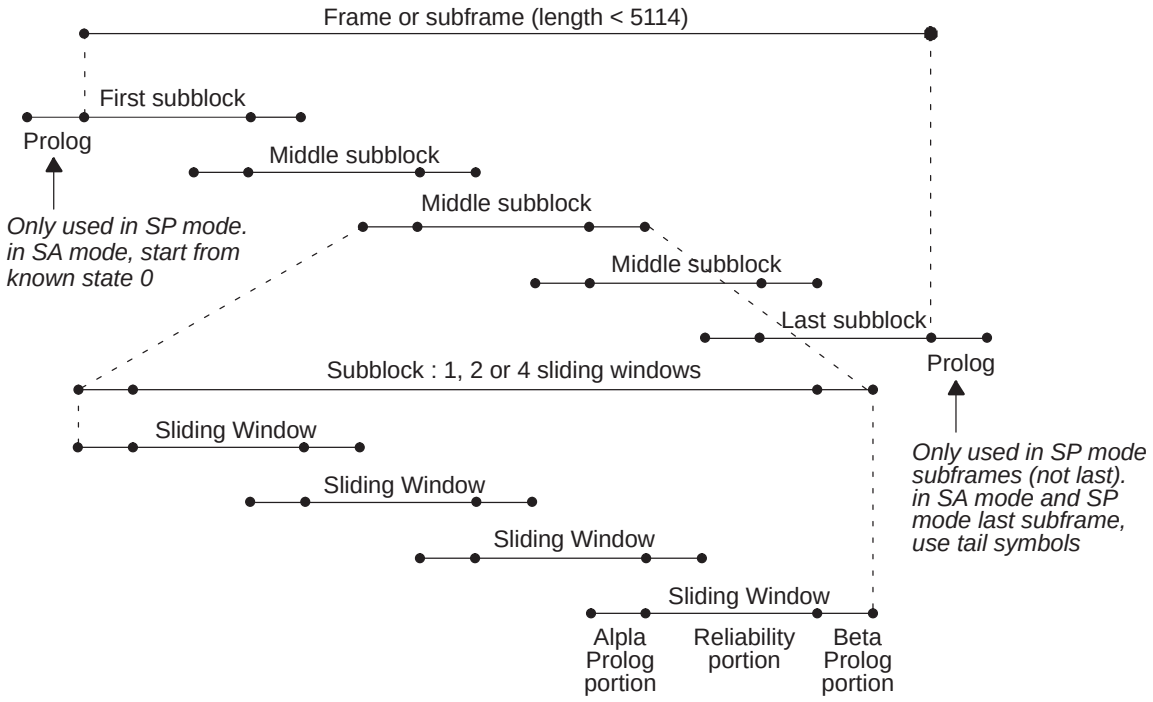


Figure 96 show diagrams for the sliding windows for both alpha and beta. Each sliding window consists of a reliability section and a prolog section. The size of the reliability section can range from 40 to 128 and the size of the prolog section can range from 4 to 48 in the TCP2. The alpha sliding window starts at the left and goes right (forward order) and the beta sliding window starts at the right and goes left (reverse order). See Table 40 for the number of sub-blocks and sliding windows.

Table 40. Examples for NUM_BLOCK, NUM_SUBBLOCK, NUM_SW, and WIN_REL

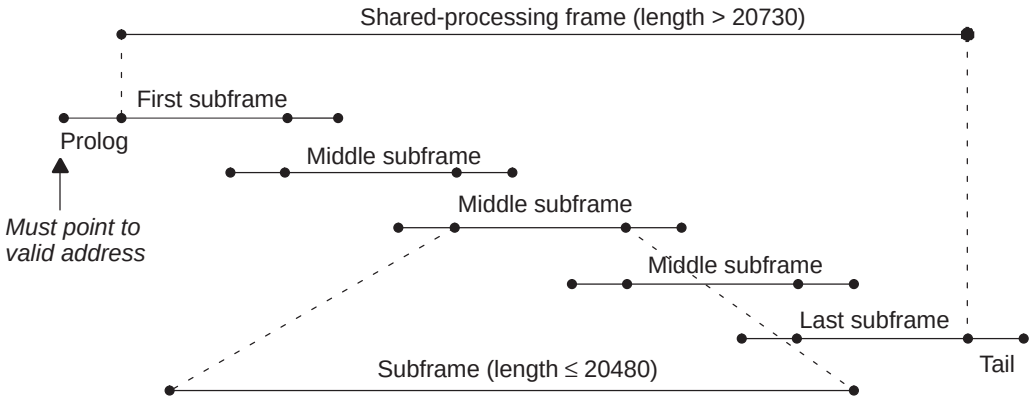
NUM_BLOCK	NUM_SUBBLOCK	NUM_SW	WIN_REL
40	1	0	40
128	1	0	128
129	1	1	65
256	1	1	128
257	2	1	65
512	2	1	128
3840	15	1	128
16122	63	1	128
20730	81	1	128

8.2 Subframe Segmentation (SP mode only)

A frame needs to be segmented into subframes when working in shared processing (SP) mode; therefore, the minimum number of subframes is 2. The subframe length should be chosen so that it has a reliable portion that is both less than 20480 and also a multiple of 256. Figure 97 shows the segmentation.

Each subframe is then further divided into sub-blocks and sliding windows as described in Section 8.1. All subframes except the last one must have the same length and be a multiple of 256.

Figure 97. Shared Processing Subframe Segmentation (Example with 5 Subframes)



8.3 Reliability and Prolog Length Calculation

F: Frame length (number of bits in a frame prior to turbo-encoding)

R: 1/code rate

P: Prolog length (number of symbols to be used in the prolog not taking into account the rate)

The unit is designed so that reliability size ranges from 40 to 128 bits and prolog size ranges from 4 to 48 bits. The prolog size is chosen based on whether the code is punctured or non-punctured. The prolog size can be programmed from 4 to 48 in the TCP2. If prolog reduction is enabled, the prolog size should range between 4 to16.

Note: In shared-processing mode, the prolog size must be a multiple of 8 due to EDMA3 transfers alignments constraints.

The reliability size is chosen to optimally fill the pipelines; however, there are some limitations. The maximum size of each sub-block is 256 symbols in which each of the two sliding windows maximum reliability size is 128. The reliability length is computed from the frame length.

Given N the block size, P the prolog length, Nsb the number of sub-blocks, Nsw the number of sliding windows per sub-block (Nsw =1 or 2), the reliability length R must fill the following properties:

- The last sliding window reliability can be smaller than the others (last beta prolog is not processed, tails are used to initialize beta states).
- The prolog of the sliding window before the last must point before the end of the frame: for example, N, P, R are such that N = (Nsb * Nsw - 1) * R + P + r; r > 0. The following formula meets the above properties:

Figure 98. Example R Formula

$R_{MAX} = 128$

if$(N \leq 128)$

$NSW = 1, R = N$

$ELSE$

$NSW = 2$

$IF(N_{SW} = 2)\{$

$while((N_{SB} \times R \times N_{SW} - N) \geq (R - 48))\{$

$WIN\_SIZE = CEIL[N/N_{SW}]$

$$N_{SB} = CEIL\left[\frac{WIN\_SIZE}{R_{MAX}}\right]$$

$$R = \frac{WIN\_SIZE}{N_{SB}}$$

$IF(R \times N_{SB} < WIN\_SIZE)R++$

$R_{MAX} = R_{MAX} - 1$

$\}$

$N_{SW} = N_{SW} - 1$

This computation should be done by the DSP CPU. It should be noted that a 1 must be subtracted from the calculated R value prior to writing to TCPIC1.

Note: The reliability length must fill the above properties to receive a correct decoding. If these rules are not followed, the MAP will execute, but the BER might not be optimal.

8.4 Added Features

8.4.1 Code Rates

TCP only supports the code rates used in the 3GPP and cdma2000 standards, namely 1/2, 1/3 and 1/4. TCP2 introduces a more flexible design which always assumes that the code rate is 1/5. Therefore, all puncturing schemes derived from the rate 1/5 mother code are automatically supported. The received systematic and parity data is first de-punctured to rate 1/5, then formatted according to the TCP specifications (see Section 4.1), and finally sent to the TCP.

8.4.2 Input Sign

The TCP assumes that the encoded bits are converted into signed binary symbols using the following mapping: $0 \rightarrow -1$, $1 \rightarrow +1$ and scaled by $-2*a/\Sigma^2$ where a is the fading factor and Σ is the noise variance. Many receivers may perform this scaling without applying the -1 factor. With TCP, this requires the DSP to perform the -1 multiplication as the TCP expects this scaling. In TCP2, the -1 multiplication can be performed automatically by the TCP2 based on the INPUTSIGN bit field of the TCPIC3 configuration register. This reduces DSP overhead and does not cost extra TCP2 cycles.

8.4.3 Log Equation

The exact mathematical equation for forward and backward recursion in a MAP decoder is based on a log of a sum of two exponential terms, $\ln(e^A + e^B)$. Due to the complexity of hardware implementation, this expression is often approximated with the co-called max* term, which is computed as max* (A, B) = max (A, B) + $\ln(1 + e^{|-A-B|})$; i.e., the maximum of the exponents and a correction term which is a function of the difference of the exponents. The correction term is often implemented as a table look-up. Such decoder is called Max* Log-MAP. If the correction term is dropped, the implementation becomes max-log-MAP.

While TCP uses a Max*-log-MAP implementation, TCP2 offers both Max* Log-MAP and max-log-MAP implementations. This can be selected on a frame-by-frame basis. The second implementation does not require the input LLRs (log-likelihood ratios) to be scaled by a factor inversely proportional to noise variance, and is therefore more robust in situations where SNR can not be accurately estimated.

8.4.4 Re-Encode

The re-encode block is directly connected to the CRC block. During the sub-block execution, up to 256 sets of data will be stored in a double buffered memory. Two bits each will be stored for x0, p0, and p1. One bit is the sign bit and the other bit is set if the symbol is equal to a zero. These 6 bits will be used for re-encoding. The seventh bit will be the hard decision bit. This bit is the sign of the following summation: (x+a+w).

The decision bits can be re-encoded with a convolutional encoder. The output of the encoder is 3 bit streams: systematic bit and 2 parity bits. These bits can be compared with the signs of the stored systematic and parity symbols. If the bits match, then no error has occurred. If the bits do not match, then an error has occurred. If the stored symbol is a zero, then no information can be determined from this data. A zero symbol represents either a depunctured symbol or a symbol that is equal distance from the ideal modulated +1 or -1. As the signs are compared, a running count of the total number of sign differences is calculated. These counts can be used as an estimate of the channel quality.

The cnt_re_map0 output register is a sum of the number of sign differences for MAP0. The cnt_re_map1 output register is a sum of the number of sign differences for MAP1. Table 41 shows the valid symbols that can be used during the calculation of errors for each MAP.

Table 41. Valid Re-Encode Symbols Used for Comparison

MAP	Valid Systematic (x)	Valid Parity (p0)	Valid Parity (p1)
0	Yes	Yes	Yes
1	No	Yes	Yes

9 Programming

The TCP2 requires setting up the following context per user channel:

- Standalone (SA) mode
  - 3 to 5 EDMA3 parameters (see Table 42)
  - The input configurations parameters
- Shared-processing (SP) mode
  - 3 to 4 EDMA3 parameters (see Table 43)
  - The input configurations parameters

Note that several user channels can be programmed prior to starting the TCP2.

Table 42. EDMA3 Parameters in Standalone (SA) Mode

Direction Transmit (DSP → TCP) Receive (TCP → DSP)	Data	Usage	Required/Optional
Transmit	Input configuration parameters	Send the input configuration parameters	Required
Transmit	Systematics and parities	Send the systematics and parities	Required
Transmit	Interleaver indexes	Send the interleaver table	Optional INTER bit
Receive	Hard decisions	Read the hard decisions	Required
Receive	Output parameters	Read the output parameters	Optional OUTF bit

Table 43. EDMA3 Parameters in Shared Processing (SP) Mode

Direction Transmit (DSP → TCP) Receive (TCP → DSP)	Data	Usage	Required/Optional
Transmit	Input configuration parameters	Send the input configuration parameters	Required
Transmit	Systematics and parities	Send the systematics and parities	Required
Transmit	A prioris	Send the a prioris	Required except for first iteration MAP1
Receive	Extrinsics	Read the extrinsics	Required

9.1 EDMA3 Resources

9.1.1 TCP2 Dedicated EDMA3 Resources

Within the available 64 EDMA3 channels event sources, two are assigned to the TCP: event 30 and event 31.

- Event 30 is associated to the TCP2 receive event (TCPREVT) and is used as the synchronization event for the EDMA3 transfer from the TCP2 to the DSP (receive).
- Event 31 is associated to the TCP2 transmit event (TCPXEVT) and is used as the synchronization event for the EDMA3 transfer from the DSP to the TCP2 (transmit).

9.1.2 Special TCP2 EDMA3 Programming Considerations

The EDMA3 parameters consists of eight words as shown in Figure 99. In the context of the TCP2, all EDMA3 transfers should follow ACNT * BCNT number of bytes in A_Sync Mode and ACNT *BCNT * CCNT number of bytes in AB_Sync Mode. The number of bytes should be a multiple of 8. For more information, see the *TMS320TCI648x DSP Enhanced Direct Memory Access (EDMA3) Controller Reference Guide* (SPRU727).

Figure 99. EDMA3 Parameters Structure

31 0

EDMA3 Channel Options Parameter (OPT)	
EDMA3 Channel Source Address (SRC)	
Number of arrays of length ACNT (BCNT)	Number of bytes in array (ACNT)
EDMA3 Channel Destination Address (DST)	
Destination 2nd Dimension Index (DSTBIDX)	Source 2nd Dimension Index (SRCBIDX)
BCNTRLD	LINK
Destination 3rd Dimension Index (DSTCIDX)	Source 3rd Dimension Index (SRCCIDX)
Reserved	Number of frames in block (CCNT)

9.2 Programming Standalone (SA) Mode

Table 42 highlights the required EDMA3 resources to perform a standalone (SA) mode decoding. Each set of EDMA3 parameters uses the EDMA3 linking capabilities. Section 9.2.1 details the EDMA3 transfers programming and Section 9.2.2 details the input parameters programming. Any notification mechanism to flag that a user-channel has just been decoded is left to you. Suggested implementation is to use the EDMA3 interrupt generation capabilities [see the *TMS320TCI648x DSP Enhanced Direct Memory Access (EDMA3) Controller Reference Guide* (SPRU727)] and program the EDMA3 to generate an interrupt after the user-channel's last TCPREVT synchronized EDMA3 transfer has completed.

9.2.1 EDMA3 Programming

9.2.1.1 Input Configuration Parameters Transfer

This EDMA3 transfer to the input configuration parameters is a 16-word TCPXEVT frame synchronized transfer. The parameters should be set as:

- OPTIONS
  - ITCCEN = 0 (Intermediate transfer complete chaining is disabled)
  - TCCEN = 0 (Transfer complete chaining is disabled)
  - ITCINTEN = 0 (Intermediate transfer complete interrupt is disabled)
  - TCINTEN = 0 (Transfer complete interrupt is disabled)
  - TCC = 1 to 63 (Transfer Complete Code)
  - TCCMODE = 0 (Normal Completion)
  - FWID = Don't care
  - STAT = 0 (Entry is updated as normal)
  - SYNCDIM = 0 (A-Sync. Each event triggers the transfer of ACNT elements.)
  - DAM = 0 (Dst addressing within an array increments. Dst is not a FIFO.)
  - SAM = 0 (Src addressing within an array increments. Source is not a FIFO.)
- SOURCE ADDRESS: User input configuration parameters start address (must be double-word aligned)
- ACNT = 64 (No of bytes in an array - 16 32 bit Input Configurations to be transferred)
- BCNT = 1 (No of arrays of length ACNT)
- DESTINATION ADDRESS: TCPIC0 (5000 0000h)
- SRCBIDX = 0 (Source 2nd Dimension Index)
- DSTBIDX = 0 (Destination 2nd Dimension Index)
- SRCCIDX = 0 (Source 3rd Dimension Index)
- DSTCIDX = 0 (Destination 3rd Dimension Index)
- CCNT = 1 (No of frames in a block)
- BCNTRLD: Don't care
- LINK ADDRESS: Address in the EDMA3 PaRAM of the EDMA3 parameters associated with the systematics and parities

Upon completion, this EDMA3 transfer is linked to the EDMA3 for systematics and parities transfer parameters.

9.2.1.2 Systematics and Parities Transfer

This EDMA3 transfer to the systematics and parities memory is a TCPXEVT frame-synchronized transfer. The parameters should be set as:

- OPTIONS:
  - ITCCEN = 0 (Intermediate transfer complete chaining is disabled)
  - TCCEN = 0 (Transfer complete chaining is disabled)
  - ITCINTEN = 0 (Intermediate transfer complete interrupt is disabled)

  - TCINTEN = 0 (Transfer complete interrupt is disabled)
  - TCC = 1 to 63 (Transfer Complete Code)
  - TCCMODE = 0 (Normal Completion)
  - FWID = Don't care
  - STAT = 0 (Entry is updated as normal)
  - SYNCDIM = 0 (AB-Sync, Each event triggers the transfer of BCNT arrays of ACNT elements.)
  - DAM = 0 (Dst addressing within an array increments. Dst is not a FIFO.)
  - SAM = 0 (Src addressing within an array increments. Source is not a FIFO.)
- SOURCE ADDRESS: Systematics and parities start address (must be double-word aligned)
- ACNT = 8*ceil(frame_length/2)
- BCNT = 1 (No of arrays of length ACNT)
- DESTINATION ADDRESS: TCPSP (5001 0000h)
- SRCBIDX = 0 (Source 2nd Dimension Index)
- DSTBIDX = 0 (Destination 2nd Dimension Index)
- SRCCIDX = 0 (Source 3rd Dimension Index)
- DSTCIDX = 0 (Destination 3rd Dimension Index)
- CCNT = 1 (No of frames in a block)
- BCNTRLD: Don't care
- LINK ADDRESS: Address in the EDMA3 PaRAM of the EDMA3 parameters associated with the systematics and parities.

1. The EDMA3 interleaver table transfer parameters, if there is a new one to be loaded in the TCP2 (INTER bit is set)
2. The EDMA3 input configuration parameters transfer parameters of the next user-channel, if there is one ready to be decoded and no interleaver table to be loaded in the TCP2 (INTER bit is cleared)
3. Dummy DMA transfer parameters, if there are no more user channels ready to be decoded (see *TMS320TCI648x DSP Enhanced Direct Memory Access (EDMA3) Controller Reference Guide* (SPRU727) for information on how to setup a dummy Xfer). Do not link to a NULL transfer, as the secondary event register sets the event flag for Event 29. The final TCPXEVT is generated upon the reading of the decisions and output registers, which is intended to transfer the input configuration of the next user channel. If a NULL transfer link is in place, the final TCPXEVT will set the event 29 flag of SER and no further TCP execution will occur until it is cleared.

9.2.1.3 Interleaver Indexes Transfer

This EDMA3 transfer to the interleaver memory is a TCPXEVT frame-synchronized transfer. The parameters should be set as:

- OPTIONS:
  - ITCCEN = 0 (Intermediate transfer complete chaining is disabled)
  - TCCEN = 0 (Transfer complete chaining is disabled)
  - ITCINTEN = 0 (Intermediate transfer complete interrupt is disabled)
  - TCINTEN = 1 (Transfer complete interrupt is enabled)
  - TCC = 1 to 63 (Transfer Complete Code)
  - TCCMODE = 0 (Normal Completion)
  - FWID = Don't care
  - STAT = 0 (Entry is updated as normal)
  - SYNCDIM = 0 (A-Sync. Each event triggers the transfer of ACNT elements.)
  - DAM = 0 (Dst addressing within an array increments. Dst is not a FIFO.)
  - SAM = 0 (Src addressing within an array increments. Source is not a FIFO.)
- SOURCE ADDRESS: Interleaver table start address (must be double-word aligned)
- ACNT = 8 * ceil ((frame_length+3)/4)) ⇒ (No of bytes in an array)
- BCNT = 1 (No of arrays of length ACNT)
- DESTINATION ADDRESS: TCPINTER (5005 0000h)

- SRCBIDX = 0 (Source 2nd Dimension Index)
- DSTBIDX = 0 (Destination 2nd Dimension Index
- SRCCIDX = 0 (Source 3rd Dimension Index)
- DSTCIDX = 0 (Destination 3rd Dimension Index)
- CCNT = 1 (No of frames in a block)
- BCNTRLD: Don't care
- LINK ADDRESS: See cases 1 and 2 below

Upon completion, this EDMA3 transfer is linked to one of the following:

1. The EDMA3 input configuration parameters transfer parameters of the next user-channel, if there is one ready to be decoded
2. Dummy DMA transfer parameters, if there are no more user channels ready to be decoded (see *TMS320TCI648x DSP Enhanced Direct Memory Access (EDMA3) Controller Reference Guide* (SPRU727) for information on how to setup a dummy Xfer). Do not link to a NULL transfer, as the secondary event register sets the event flag for Event 29. The final TCPXEVT is generated upon the reading of the decisions and output registers, which is intended to transfer the input configuration of the next user channel. If a NULL transfer link is in place, the final TCPXEVT will set the event 29 flag of SER and no further TCP execution will occur until it is cleared.

9.2.1.4 Hard-Decisions Transfer

This EDMA3 transfer to the hard decisions buffer is a TCPREVT frame-synchronized transfer. The parameters should be set as:

- OPTIONS:
  - ITCCEN = 0 (Intermediate transfer complete chaining is disabled)
  - TCCEN = 0 (Transfer complete chaining is disabled)
  - ITCINTEN = 0 (Intermediate transfer complete interrupt is disabled)
  - TCINTEN = 0 (Transfer complete interrupt is disabled)
  - TCC = 1 to 63 (Transfer Complete Code)
  - TCCMODE = 0 (Normal Completion)
  - FWID = Don't care
  - STAT = 0 (Entry is updated as normal)
  - SYNCDIM = 0 (A-Sync. Each event triggers the transfer of ACNT elements.)
  - DAM = 0 (Dst addressing within an array increments. Dst is not a FIFO.)
  - SAM = 0 (Src addressing within an array increments. Source is not a FIFO.)
- SOURCE ADDRESS: TCPHD (5006 0000h)
- ACNT = 8 * ceil (frame_length/64) ⇒ (No of bytes in an array. Note that this implies that the destination location must have 8*ceil(frame_length/64) bytes allocated for decisions.)
- BCNT = 1 (No of arrays of length ACNT)
- DESTINATION ADDRESS: User hard decisions start address (must be double-word aligned)
- ELEMENT INDEX: Don't care
- SRCBIDX = 0 (Source 2nd Dimension Index)
- DSTBIDX = 0 (Destination 2nd Dimension Index)
- SRCCIDX = 0 (Source 3rd Dimension Index)
- DSTCIDX = 0 (Destination 3rd Dimension Index)
- CCNT = 1 (No of frames in a block)
- BCNTRLD: Don't care
- LINK ADDRESS: See cases 1, 2, and 3 below

Upon completion, this EDMA3 transfer is linked to one of the following:

1. The EDMA3 hard decisions transfer parameters of the next user-channel, if there is one ready to be decoded and the OUTF bit is cleared.
2. The EDMA3 output parameters transfer parameters, if the OUTF bit is set.

3. Null EDMA3 transfer parameters (with all zeros), if there are no more user channels ready to be decoded and the OUTF bit is cleared.

9.2.1.5 Output Parameters Transfer

This EDMA3 transfer is optional and depends on the OUTF bit in the TCP2 input configuration register 0 (TCPIC0). This EDMA3 transfer is a TCPREVT frame-synchronized transfer. The parameters should be set as:

- OPTIONS:
  - ITCCEN = 0 (Intermediate transfer complete chaining is disabled)
  - TCCEN = 0 (Transfer complete chaining is disabled)
  - ITCINTEN = 0 (Intermediate transfer complete interrupt is disabled)
  - TCINTEN = 1 (Transfer complete interrupt is enabled)
  - TCC = 1 to 63 (Transfer Complete Code)
  - TCCMODE = 0 (Normal Completion)
  - FWID = Don't care
  - STAT = 0 (Entry is updated as normal)
  - SYNCDIM = 0 (A-Sync. Each event triggers the transfer of ACNT elements.)
  - DAM = 0 (Dst addressing within an array increments. Dst is not a FIFO)
  - SAM = 0 (Src addressing within an array increments. Source is not a FIFO.)
- SOURCE ADDRESS: TCPOUT (5000 0030h)
- ACNT = 12 (No of bytes in an array)
- BCNT = 1 (No of arrays of length ACNT)
- DESTINATION ADDRESS: User output parameters start address (must be double-word aligned)
- SRCBIDX = 0 (Source 2nd Dimension Index)
- DSTBIDX = 0 (Destination 2nd Dimension Index)
- SRCCIDX = 0 (Source 3rd Dimension Index)
- DSTCIDX = 0 (Destination 3rd Dimension Index)
- CCNT = 1 (No of frames in a block)
- BCNTRLD: Don't care
- LINK ADDRESS: See cases 1 and 2 below

Upon completion, this EDMA3 transfer is linked to one of the following:

1. The hard decisions EDMA3 transfer parameters of the next user-channel, if there is one ready to be decoded.
2. Null EDMA3 transfer parameters (with all zeros), if there are no more user-channels ready to be decoded.

9.2.2 Input Configurations Parameters Programming

The frame length (FL bits in TCPIC0) should be set to the number of information bits (prior to turbo-encoding and not including the rate or the tail bits).

The number of sub-blocks (NSB bits in TCPIC2), the reliability length (R bits in TCPIC1), and the prolog size (P bits in TCPIC2) should be set as described in Section 8.3. It should be noted that a 1 must be subtracted from the calculated R value prior to writing to TCPIC1.

The maximum number of iterations (MAXIT bits in TCPIC2) should be selected as a function of the overall system performance. A value 0 sets the maximum number of iterations to its maximum (32).

The SNR threshold ratio (SNR bits in TCPIC2) should be selected as a function of the overall system performance. A value 0 disables the stopping criteria algorithm.

The EMAXSTR bit can be enabled or disabled in TCPIC3 (0 = max star disabled (enable Max Log-MAP) 1 = max star enabled (enable log MAP)).

The minimum number of iterations (MINIT bits in TCPIC3) should be selected as a function of the overall system performance (minimum iterations 1 to 31) when SNR stopping criteria is used.

The INPUTSIGN bit can be enabled or disabled in TCPIC3 (0 = Use channel input data as is, 1 = multiply channel input data by -1).

The OUTORDER bit can be enabled or disabled in TCPIC3 (0 = output bit ordering from 0 to 31, 1 = output bit ordering from 31 to 0).

The EPRORED bit can be enabled or disabled in TCPIC3 (0 = prolog reduction disabled, 1 = prolog reduction enabled).

The CRC length and CRC iterations (TCPIC4) and CRC Polyn bits (TCPIC5) should be selected as a function of the overall system performance. A value 0 disables the CRC stopping criteria algorithm.

The TAIL1, TAIL2, TAIL3, TAIL4, TAIL5, and TAIL6 bits should be programmed as described in Section 6.8 through Section 6.13, respectively.

The Extrinsic Scaling factors can be selected in registers TCPIC12, TCPIC13, TCPIC14, and TCPIC15.

Table 44. Input Configuration Parameters Settings in Standalone (SA) Mode

Bit Field	Register	Value
OPMOD	TCPIC0	OPMOD = 00: SA Mode
INTER	TCPIC0	INTER = 0 if no new interleaver table is needed; otherwise, INTER = 1
OUTF	TCPIC0	OUTF = 1 if TCPREVT is to be generated for the output parameters load; otherwise, OUTF = 0

9.3 Programming Shared-Processing (SP) Mode

In shared mode, the DSP must do more work and work closely with the TCP2. The DSP breaks the large frame into smaller frames of 20,480 or less. Each one of these frames is called a subframe. The size of all the subframes (except the last subframe) must be divisible by 256. Note that the frame_length listed in the shared-processing mode is the length of the subframes and not the length of the frame. The TCP will treat each subframe as its own frame of data.

To decode the whole frame, follow these steps:

1. DSP sends subframe systematic, parity and extrinsic data to TCP2.
2. TCP2 executes two MAP decoders for each iteration.
3. DSP reads the intermediate results (extrinsics).
4. DSP interleaves or de-interleaves data.
5. Steps 1 to 4 are repeated for all subframes.

The opmod parameter defines which subframe the TCP2 is decoding. Opmode is set to 1 for the first subframe, opmode is set to 2 for the middle subframe(s), and opmode is set to 3 for the last subframe.

Table 43 highlights the required EDMA3 resources to perform a shared-processing (SP) mode decoding. As in standalone (SA) mode decoding, each set of EDMA3 parameters uses the EDMA3 linking capabilities. In addition, the a priori data transfer is done using the EDMA3 alternate transfer chaining capabilities. Section 9.3.1 details the EDMA3 transfers programming and Section 9.3.2 details the input parameters programming. It should be noted that any stopping criteria algorithm has to be implemented by the CPU.

Any notification mechanism to flag that a user-channel has just been decoded is left to you. Suggested implementation is to use the EDMA3 interrupt generation capabilities [see *TMS320TCI648x DSP Enhanced Direct Memory Access (EDMA3) Controller Reference Guide* (SPRU727)] and program the EDMA3 to generate an interrupt after the user-channel's last TCPREVT synchronized EDMA3 transfer has completed.

9.3.1 EDMA3 Programming

9.3.1.1 Input Configuration Parameters Transfer

This EDMA3 transfer to the input configuration parameters is a 16-word TCPXEVT frame-synchronized transfer. The parameters should be set as:

- OPTIONS:
  - ITCCEN = 0 (Intermediate transfer complete chaining is disabled)
  - TCCEN = 0 (Transfer complete chaining is disabled)
  - ITCINTEN = 0 (Intermediate transfer complete interrupt is disabled)
  - TCINTEN = 0 (Transfer complete interrupt is disabled)
  - TCC = 1 to 63 (Transfer Complete Code)
  - TCCMODE = 0 (Normal Completion)
  - FWID = Don't care
  - STAT = 0 (Entry is updated as normal)
  - SYNCDIM = 0 (A-Sync. Each event triggers the transfer of ACNT elements.)
  - DAM = 0 (Dst addressing within an array increments. Dst is not a FIFO.)
  - SAM = 0 (Src addressing within an array increments. Source is not a FIFO.)
- SOURCE ADDRESS: user input configuration parameters start address (must be double-word aligned)
- ACNT = 64 (No of bytes in an array)
- BCNT = 1 (No of arrays of length ACNT)
- DESTINATION ADDRESS: TCPIC0 (5000 0000h)
- SRCBIDX = 0 (Source 2nd Dimension Index)
- DSTBIDX = 0 (Destination 2nd Dimension Index)
- SRCCIDX = 0 (Source 3rd Dimension Index)
- DSTCIDX = 0 (Destination 3rd Dimension Index)
- CCNT = 1 (No of frames in a block)
- BCNTRLD: Don't care
- LINK ADDRESS: Address in the EDMA3 PaRAM of the EDMA3 parameters associated with the systematics and parities.

Upon completion, this EDMA3 transfer is linked to the EDMA3 for systematics and parities transfer parameters.

9.3.1.2 Systematics and Parities Transfer

This EDMA3 transfer to the systematics and parities memory is a TCPXEVT frame-synchronized transfer. The parameters should be set as:

- OPTIONS:
  - ITCCEN = 0 (Intermediate transfer complete chaining is disabled)
  - TCCEN = 0 (Transfer complete chaining is disabled)
  - ITCINTEN = 0 (Intermediate transfer complete interrupt is disabled)
  - TCINTEN = 0 (Transfer complete interrupt is disabled)
  - TCC = 1 to 63 (Transfer Complete Code)
  - TCCMODE = 0 (Normal Completion)
  - FWID = Don't care
  - STAT = 0 (Entry is updated as normal)
  - SYNCDIM = 1 (AB-Sync. Each event triggers the transfer of BCNT arrays of ACNT elements.)
  - DAM = 0 (Dst addressing within an array increments. Dst is not a FIFO.)
  - SAM = 0 (Src addressing within an array increments. Source is not a FIFO.)
- SOURCE ADDRESS: Systematics and parities start address (must be double-word aligned)
- ACNT = 8 (No of bytes in an Array)

- Word count = 2 * ceil (frame_length/2)
- BCNT = (Word count /2) (No of arrays of length ACNT)
- DESTINATION ADDRESS: TCPSP (5001 0000h)
- SRCBIDX = 8 (Source 2nd Dimension Index)
- DSTBIDX = 8 (Destination 2nd Dimension Index)
- SRCCIDX = 8 (Source 3rd Dimension Index)
- DSTCIDX = 8 (Destination 3rd Dimension Index)
- CCNT = 8 (No of frames in a block)
- BCNTRLD: Don't care
- LINK ADDRESS: See cases 1 and 2 below

Upon completion, this EDMA3 transfer is linked to one of the following:

1. The EDMA3 input configuration parameters transfer parameters of the next user-channel, if there is one ready to be decoded and the current decoding is a MAP0 from the first iteration.
2. Dummy EDMA3 transfer parameters, if there are no more user channels ready to be decoded.

9.3.1.3 A Priori Transfer

This EDMA3 transfer to the a priori memory is a TCPXEVT chained and frame-synchronized transfer. This EDMA3 transfer is chained from the systematic and parity data transfer and occurs only when executing any MAP but the MAP0 of the first iteration; that is, the OPMOD bits in TCPIC0 must be set to 2h, 4h, and 6h respectively. The parameters should be set as:

- OPTIONS:
  - ITCCEN = 0 (Intermediate transfer complete chaining is disabled)
  - TCCEN = 0 (Transfer complete chaining is disabled)
  - ITCINTEN = 0 (Intermediate transfer complete interrupt is disabled)
  - TCINTEN = 1 (Transfer complete interrupt is enabled)
  - TCC = 1 to 63 (Transfer Complete Code)
  - TCCMODE = 0 (Normal Completion)
  - FWID = Don't care
  - STAT = 0 (Entry is updated as normal)
  - SYNCDIM = 1 (A-Sync. Each event triggers the transfer of ACNT elements.)
  - DAM = 0 (Dst addressing within an array increments. Dst is not a FIFO.)
  - SAM = 0 (Src addressing within an array increments. Source is not a FIFO.)
- SOURCE ADDRESS: A priori start address (must be double-word aligned)
- If the OPMOD == FIRST_SUB_FRAME
  ACNT = 8 * ceil ((frame_length + prolog_length)/8) ⇒ (No of bytes in an array)
- If the OPMOD == MIDDLE_SUB_FRAME
  ACNT = 8 * ceil ((frame_length + 2 * prolog_length)/8) ⇒ (No of bytes in an array)
- If the OPMOD == LAST_SUB_FRAME
  ACNT = 8 * ceil ((frame_length + prolog_length)/8) ⇒ (No of bytes in an array)
- BCNT = 1 (No of arrays of length ACNT)
- DESTINATION ADDRESS: TCPAP (5004 0000h)
- SRCBIDX = 0 (Source 2nd Dimension Index)
- DSTBIDX = 0 (Destination 2nd Dimension Index)
- SRCCIDX = 0 (Source 3rd Dimension Index)
- DSTCIDX = 0 (Destination 3rd Dimension Index)
- CCNT = 1 (No of frames in a block)
- BCNTRLD: Don't care
- LINK ADDRESS: See cases 1 and 2 below

Upon completion, this EDMA3 transfer is linked to one of the following:

1. The EDMA3 input configuration parameters transfer parameters of the next user-channel MAP, if there is one ready to be decoded.
2. Dummy EDMA3 transfer parameters, if there are no more user channels LOG-MAP ready to be decoded.

9.3.1.4 Extrinsics Transfer

This EDMA3 transfer to the extrinsics buffer is a TCPREVT frame-synchronized transfer. The parameters should be set as:

- OPTIONS:
  - ITCCEN = 0 (Intermediate transfer complete chaining is disabled)
  - TCCEN = 0 (Transfer complete chaining is disabled)
  - ITCINTEN = 0 (Intermediate transfer complete interrupt is disabled)
  - TCINTEN = 1 (Transfer complete interrupt is disabled)
  - TCC = 1 to 63 (Transfer Complete Code)
  - TCCMODE = 0 (Normal Completion)
  - FWID = Don't care
  - STAT = 0 (Entry is updated as normal)
  - SYNCDIM = 0 (A-Sync. Each event triggers the transfer of ACNT elements.)
  - DAM = 0 (Dst addressing within an array increments. Dst is not a FIFO.)
  - SAM = 0 (Src addressing within an array increments. Source is not a FIFO.)
- If the OPMODE == FIRST_SUB_FRAME
  SOURCE ADDRESS: TCPEXT (5003 0000h)
- If the OPMODE == MIDDLE_SUB_FRAME or LAST_SUB_FRAME
  SOURCE ADDRESS: TCPEXT (5003 0000h + prolog_length)
- ACNT = 8 * ceil ((frame_length)/8) ⇒ (No of bytes in an array)
- BCNT = 1 (No of arrays of length ACNT)
- DESTINATION ADDRESS: User extrinsics start address (must be double-word aligned)
- SRCBIDX = 0 (Source 2nd Dimension Index)
- DSTBIDX = 0 (Destination 2nd Dimension Index)
- SRCCIDX = 0 (Source 3rd Dimension Index)
- DSTCIDX = 0 (Destination 3rd Dimension Index)
- CCNT = 1 (No of frames in a block)
- BCNTRLD: Don't care
- LINK ADDRESS: See cases 1 and 2 below

Upon completion, this EDMA3 transfer is linked to one of the following:

1. The EDMA3 extrinsics transfer parameters of the next user-channel, if there is one ready to be decoded.
2. Null EDMA3 transfer parameters (all with all zeros), if there are no more user-channels ready to be decoded.

9.3.2 Input Configurations Parameters Programming

The frame length (FL bits in TCPIC0) should be set to the total shared-processing frame length (prior to turbo-encoding and not including any tail information).

The maximum number of iterations (MAXIT bits in TCPIC2) should be selected as a function of the overall system performance. A value of 0 sets the maximum number of iterations to its maximum (32).

- The SNR threshold ratio (SNR bits in TCPIC2) should be disabled.
- The CRC bits in (TCPIC4) should be disabled.
- The prolog reduction should be disabled.
- The extrinsic scaling should be disabled.

- The EMAXSTR bit can be enabled or disabled in TCPIC3, 0 = max star disabled (enable Max Log-MAP, 1 = max star enabled (enable log MAP)).
- The minimum number of iterations (MINIT bits in TCPIC3) should be selected as a function of the overall system performance (minimum iterations 1 to 31).
- The INPUTSIGN bit can be enabled or disabled in TCPIC3 (0 = Use channel input data as is, 1 = multiply channel input data by -1).
- The OUTORDER bit can be enabled or disabled in TCPIC3 (0 = output bit ordering from 0 to 31, 1 = output bit ordering from 31 to 0).

The TAIL1, TAIL2, TAIL3, TAIL4, TAIL5, TAIL6 bits should be programmed as described in Section 6.7 through Section 6.12, respectively.

Table 45. Input Configuration Parameters Settings in Shared-Processing (SP) Mode

Bit field	Register	Value
OPMOD	TCPIC0	OPMOD = 2 for first subframe OPMOD = 4 for middle subframe OPMOD = 6 for last subframe
INTER	TCPIC0	INTER = 0
OUTF	TCPIC0	OUTF = 0

10 Output Parameters

The various output parameters are described in Section 6.19 through Section 6.21. These parameters provide useful information when the stopping criteria are enabled.

11 Events Generation

A TCP2 transmit event (TCPXEVT) and TCP2 receive event (TCPREVT) are generated as shown in Figure 100 for standalone (SA) mode and as shown in Figure 101 for shared-processing (SP) mode (example of 2 subframes).

Figure 100. TCP2 Events Generation in Standalone (SA) Mode

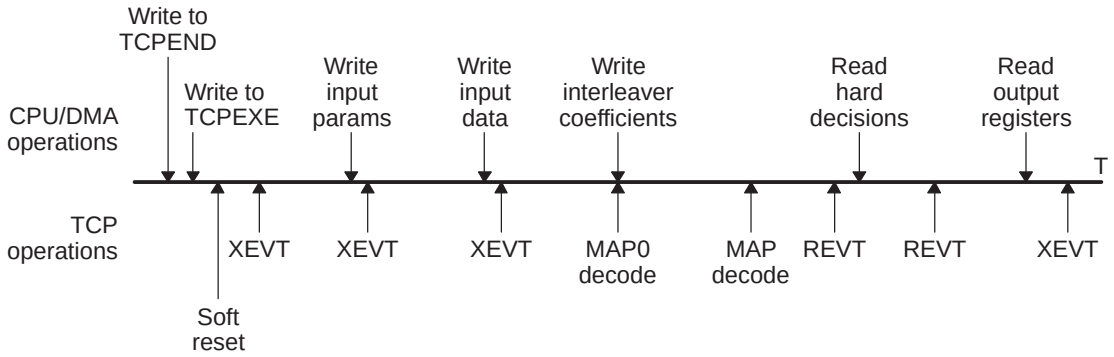
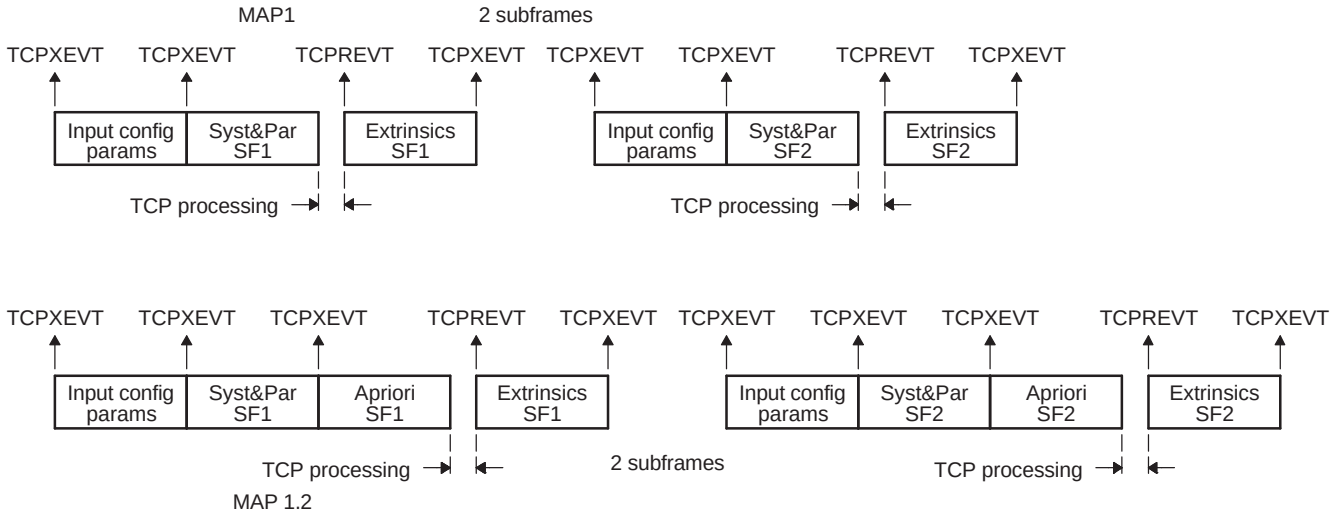


Figure 101. TCP2 Events Generation in Shared-Processing (SP) Mode



12 Debug Mode: Pause After Each Map

The TCPEXE register starts, resets, and places TCP2 into debug mode. Writing the following to TCPEXE will place TCP2 into the defined modes.

- 0 = no instruction. Value at reset or value written by the coprocessor when previous instruction is read and its execution is ongoing. DSP may test the status word in the output control memory to check if the instruction is being executed.
- 1 = start. The TCI648x CPU requests the coprocessor to start a processing block. The first action of the coprocessor is to stop any of the ongoing processing, reset all its pointers and start a new process by generating the first XEVT to trigger EDMA3 transfer of the input control words.
- 4 = debug mode. Normal initialization and wait in MAP state 0.
- 5 = debug mode. Execute one MAP decode and wait in MAP state 6.
- 6 = debug mode. Execute remaining MAP decodes and complete normal ending.
- 7 = SOFT RESET. Soft reset all TCP2 registers, except for endianness, execution, emulation register, and all other internal registers.

13 Errors and Status

13.1 Errors

The TCP2 error register (TCPERR) flags any errors that occurred in the TCP2. Once the errors are flagged, the TCP2 stops, and a TCP2_INT interrupt is generated. TCP2_INT has an interrupt selector value of 31. For details on how to set up interrupts, see the *TMS320C64x+ Megamodule Reference Guide* (SPRU871).

Reading TCPERR resets both TCPERR and the TCP2 status register (TCPSTAT) to their default values; that is, the TCP2 waits for a new START command.

13.1.1 Error Status: ERR

The ERR bit is set to 1 in case of error.

13.1.2 Unexpected Frame Length: F

The F bit is set to 1 if the programmed frame length is strictly smaller than 40 or is strictly greater than 20730 for standalone mode.

The F bit is set to 1 if the programmed frame length has the following values for shared processing mode:

1. Frame length < 256 or frame length > 20480 or (frame length)%256 != 0 if opmode = 1 or 2.
2. Frame length < 128 or frame length > 20480 if opmode = 3.

13.1.3 Unexpected Prolog Length: P

The P bit is set to 1 if the specified prolog length is strictly greater than 48. Values smaller than 4 are ignored by the hardware and 24 is used.

13.1.4 Unexpected Subframe Length: SF

The SF bit is set to 1 if the specified subframe length is strictly greater than 20480 in shared processing mode.

13.1.5 Unexpected Reliability Length: R

The R bit is set to 1 if the specified reliability length minus 1 is strictly smaller than 40 or greater than 128 in SA mode.

The R bit is set to 1 if the specified reliability length is not equal to 128 in SP mode if opmode = 1 or 2.

The R bit is set to 1 if the specified reliability length is less than 65 in SP mode if opmode = 3.

13.1.6 Unexpected Signal to Noise Ratio: SNR

The SNR bit is set to 1 if the signal to noise ratio threshold is greater than 100.

13.1.7 Unexpected Interleaver Table Load: INT

The INT bit is set to 1 if loading an interleaver table has been requested in SP mode.

13.1.8 Unexpected Output Parameters Load: OP

The OP bit is set to 1 if loading the output parameters has been requested in SP mode.

13.1.9 Unexpected Memory Access: ACC

The ACC bit is set to 1 when an unexpected memory access occurs. This can be used to spot any EDMA3 programming issues. This can occur when:

- TCP2 is waiting for input configuration parameters state and memory access to any TCP2 memory but the interleaver memory is performed
- TCP2 is waiting for systematics and parities state and memory access to any TCP2 memory other than the TCPINTER memory
- TCP2 is waiting for a prioris state and memory access to any TCP2 memory other than the TCPAP memory
- TCP2 is waiting for extrinsics state and memory access to any TCP2 memory other than the TCPEXT memory
- TCP2 is waiting for hard decisions state and memory access to any TCP2 memory other than the TCPHD memory
- TCP2 is waiting for output parameters state and memory access to any TCP2 memory

13.1.10 Unexpected Max and Min Iterations: MAXMINITER

The MAXMINITER bit is set to 1 if the minimum iterations are greater than the maximum iterations.

13.2 Status

The TCP2 status register (TCPSTAT) reflects the state of the TCP2.

13.2.1 TCP2 Decoder Status: dec_busy

The dec_busy is set to 0 if the MAP decoder is in state 0. The dec_busy is set to 1 if the MAP decoder is in states 1 to 8.

13.2.2 TCP2 Stopped Due to Error: ERR

The ERR bit is set to 1 if the TCP2 has encountered an error. The ERR bit is reset by writing a new START command in the TCP2 execution register (TCPEXE).

13.2.3 TCP2 Waiting for Input Control Parameters Write: WIC

The WIC bit is set to 1 when the TCP2 is waiting for the input configurations parameters to be written. After the very first decoding is finished, an XEVT is generated to allow any ready user-channel to be decoded and the WIC bit is set to 1.

13.2.4 TCP2 Waiting for Interleaver Table Write: WINT

The WINT bit is set to 1 when the TCP2 is waiting for the interleaver table to be written.

13.2.5 TCP2 Waiting for Systematics and Parities Write: WSP

The WSP bit is set to 1 when the TCP2 is waiting for the systematics and parities to be written.

13.2.6 TCP2 Waiting for A prioris Write: WAP

The WAP bit is set to 1 when the TCP2 is waiting for the a prioris to be written.

13.2.7 TCP2 Waiting for Extrinsics Read: REXT

The REXT bit is set to 1 when the TCP2 is waiting for the extrinsics to be read.

13.2.8 TCP2 Waiting for Hard-Decisions Read: RHD

The RHD bit is set to 1 when the TCP2 is waiting for the hard decisions to be read.

13.2.9 TCP2 Waiting for Output Parameters Read: ROP

The ROP bit is set to 1 when the TCP2 is waiting for the output parameters to be read.

13.2.10 TCP2 Halted Due to Emulation: emuhalt

The emuhalt bit is set to 1 when the TCP2 is halted due to emulation.

13.2.11 TCP2 Active Map status: Active_map

The active_map bit is set to 1 when the TCP2 is processing MAP1.

13.2.12 TCP2 Active State Status: Active_state

The Active_state indicates active MAP decoder state.

13.2.13 TCP2 Active Iteration Status: Active_iter

The Active_iter indicates active TCP2 iteration.

13.2.14 TCP2 SNR Status: snr_exceed

The snr_exceed indicates failed or passed MAPs with respect to SNR.

13.2.15 TCP2 CRC Status: Crc_pass

The Crc_pass bit is set to 1 when the CRC has passed.

13.2.16 TCP2 State: TCP_STATE

The TCP_STATE indicates tcp top level state of state machine.

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All products are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by government requirements, testing of all parameters of each product is not necessarily performed.

TI assumes no liability for applications assistance or customer product design. Customers are responsible for their products and applications using TI components. To minimize the risks associated with customer products and applications, customers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any TI patent right, copyright, mask work right, or other TI intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information published by TI regarding third-party products or services does not constitute a license from TI to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of TI information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. Reproduction of this information with alteration is an unfair and deceptive business practice. TI is not responsible or liable for such altered documentation. Information of third parties may be subject to additional restrictions.

Resale of TI products or services with statements different from or beyond the parameters stated by TI for that product or service voids all express and any implied warranties for the associated TI product or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

TI products are not authorized for use in safety-critical applications (such as life support) where a failure of the TI product would reasonably be expected to cause severe personal injury or death, unless officers of the parties have executed an agreement specifically governing such use. Buyers represent that they have all necessary expertise in the safety and regulatory ramifications of their applications, and acknowledge and agree that they are solely responsible for all legal, regulatory and safety-related requirements concerning their products and any use of TI products in such safety-critical applications, notwithstanding any applications-related information or support that may be provided by TI. Further, Buyers must fully indemnify TI and its representatives against any damages arising out of the use of TI products in such safety-critical applications.

TI products are neither designed nor intended for use in military/aerospace applications or environments unless the TI products are specifically designated by TI as military-grade or "enhanced plastic." Only products designated by TI as military-grade meet military specifications. Buyers acknowledge and agree that any such use of TI products which TI has not designated as military-grade is solely at the Buyer's risk, and that they are solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI products are neither designed nor intended for use in automotive applications or environments unless the specific TI products are designated by TI as compliant with ISO/TS 16949 requirements. Buyers acknowledge and agree that, if they use any non-designated products in automotive applications, TI will not be responsible for any failure to meet such requirements.

Following are URLs where you can obtain information on other Texas Instruments products and application solutions:

Products		**Applications**	
Amplifiers	amplifier.ti.com	Audio	www.ti.com/audio
Data Converters	dataconverter.ti.com	Automotive	www.ti.com/automotive
DSP	dsp.ti.com	Broadband	www.ti.com/broadband
Clocks and Timers	www.ti.com/clocks	Digital Control	www.ti.com/digitalcontrol
Interface	interface.ti.com	Medical	www.ti.com/medical
Logic	logic.ti.com	Military	www.ti.com/military
Power Mgmt	power.ti.com	Optical Networking	www.ti.com/opticalnetwork
Microcontrollers	microcontroller.ti.com	Security	www.ti.com/security
RFID	www.ti-rfid.com	Telephony	www.ti.com/telephony
RF/IF and ZigBee® Solutions	www.ti.com/lprf	Video & Imaging	www.ti.com/video
		Wireless	www.ti.com/wireless

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265