TMS320C64x DSP Viterbi-Decoder Coprocessor (VCP) Reference Guide

Literature Number: SPRU533C
November 2003

Texas Instruments

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 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.

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.

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
Interface	interface.ti.com	Digital Control	www.ti.com/digitalcontrol
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
		Telephony	www.ti.com/telephony
		Video & Imaging	www.ti.com/video
		Wireless	www.ti.com/wireless

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

Preface

Read This First

About This Manual

Channel decoding of voice and low bit-rate data channels found in third generation (3G) cellular standards requires decoding of convolutional encoded data. The Viterbi-decoder coprocessor (VCP) in the TMS320C6416 digital signal processor (DSP) 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 VCP.

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.

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.

TMS320C6000 CPU and Instruction Set Reference Guide (literature number SPRU189) describes the TMS320C6000™ CPU architecture, instruction set, pipeline, and interrupts for these digital signal processors.

TMS320C6000 DSP Peripherals Overview Reference Guide (literature number SPRU190) describes the peripherals available on the TMS320C6000™ DSPs.

TMS320C64x Technical Overview (SPRU395) gives an introduction to the TMS320C64x™ DSP and discusses the application areas that are enhanced by the TMS320C64x VelociTI™.

TMS320C6000 Programmer's Guide (literature number SPRU198) describes ways to optimize C and assembly code for the TMS320C6000™ DSPs and includes application program examples.

TMS320C6000 Code Composer Studio Tutorial (literature number SPRU301) introduces the Code Composer Studio™ integrated development environment and software tools.

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

TMS320C6x Peripheral Support Library Programmer's Reference (literature number SPRU273) describes the contents of the TMS320C6000™ peripheral support library of functions and macros. It lists functions and macros both by header file and alphabetically, provides a complete description of each, and gives code examples to show how they are used.

TMS320C6000 Chip Support Library API Reference Guide (literature number SPRU401) describes a set of application programming interfaces (APIs) used to configure and control the on-chip peripherals.

Trademarks

Code Composer Studio, C6000, C62x, C64x, C67x, TMS320C6000, TMS320C62x, TMS320C64x, TMS320C67x, and VelociTI are trademarks of Texas Instruments.

Contents

Figures

Tables

Viterbi-Decoder Coprocessor (VCP)

Channel decoding of voice and low bit-rate data channels found in third generation (3G) cellular standards requires decoding of convolutional encoded data. The Viterbi-decoder coprocessor (VCP) in the TMS320C6416 digital signal processor (DSP) 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 VCP.

1 Features

The VCP provides:

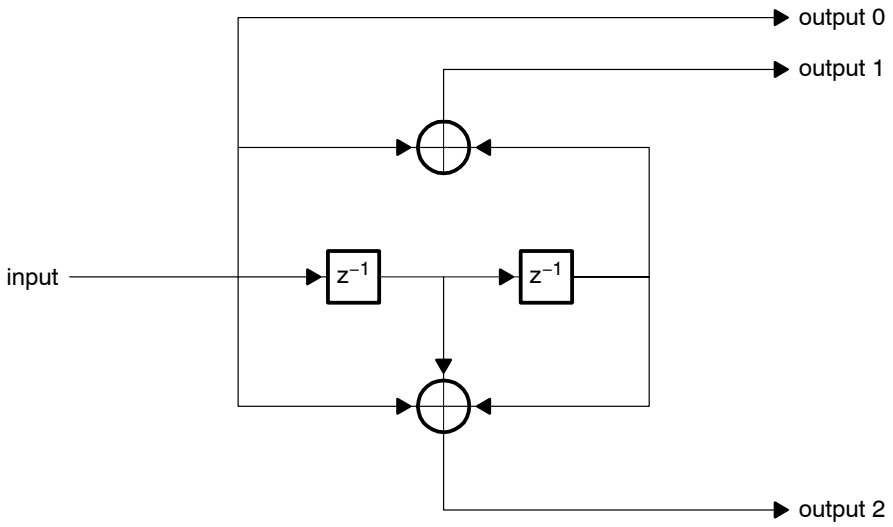
- High flexibility:
  - variable constraint length, K = 5, 6, 7, 8, or 9
  - user-supplied code coefficients
  - code rates (1/2, 1/3, or 1/4)
  - configurable trace back settings (convergence distance, frame structure)
  - branch metrics calculation and depuncturing done in software by the DSP
  - frees-up DSP resources for other processing
- System cost optimization:
  - Reduces board space and power consumption by performing decoding on-chip
  - Communication between the DSP and the VCP is performed through a high performance DMA engine
  - VCP uses its own optimized working memories

2 Introduction

A convolutional code is generated by passing the information sequence to be transmitted through a linear finite-state shift register. The VCP is able to decode only a subset of those codes known as single-shift register, nonrecursive convolutional code (an example is given in Figure 1). Important parameters for this type of codes are:

- The constraint length K (length of the delay line, the VCP supports K values from 5 to 9).
- The rate R given by $R = k/n$ where k is the number of information bits needed to produce n output bits also known as codewords (the VCP supports 1/2, 1/3, and 1/4 codes with rates).
- The generator polynomials G_n describe how the outputs are generated from the inputs.

Figure 1. Convolutional Encoder Example Block Diagram



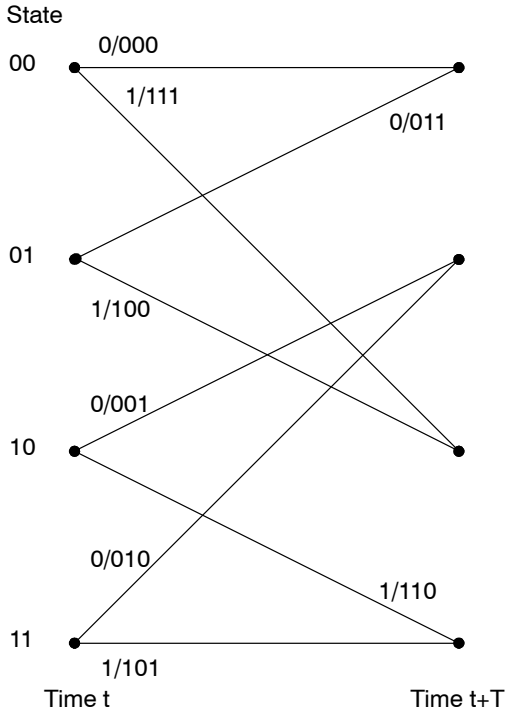
Note: K = 3, R = k/n = 1/3, $G_0 = (100)_8$, $G_1 = (101)_8$, $G_2 = (111)_8$
0/000 means input is 0, output0 is 0, output1 is 0, output2 is 0.
There are $2^{(K-1)}$ states and 2^k incoming branches per state.

From the parameters, we can derive a trellis diagram providing a useful representation of the code but whose complexity grows exponentially with the constraint length K. Figure 2 shows the trellis diagram of the code from Figure 1. The fact that there is a limited number of possible transitions from one state to another makes the code powerful and will be used in the decoding process.

As a maximum-likelihood sequence estimation (MLSE) decoder, the Viterbi decoder identifies the code sequence with the highest probability of matching the transmitted sequence based on the received sequence.

The Viterbi algorithm is composed of a metric update and a traceback routine. The metric update performs a forward recursion in the trellis over a finite number of symbol periods where probabilities are accumulated (the VCP accumulates on 12 bits) for each individual state based on the current input symbol (branch metric information). The accumulated metric is known as path metrics or state metrics. Once a path through the trellis is identified, the traceback routine performs a backward recursion in the trellis and outputs hard decisions or soft decisions.

Figure 2. Trellis Diagram for Convolutional Encoder Example



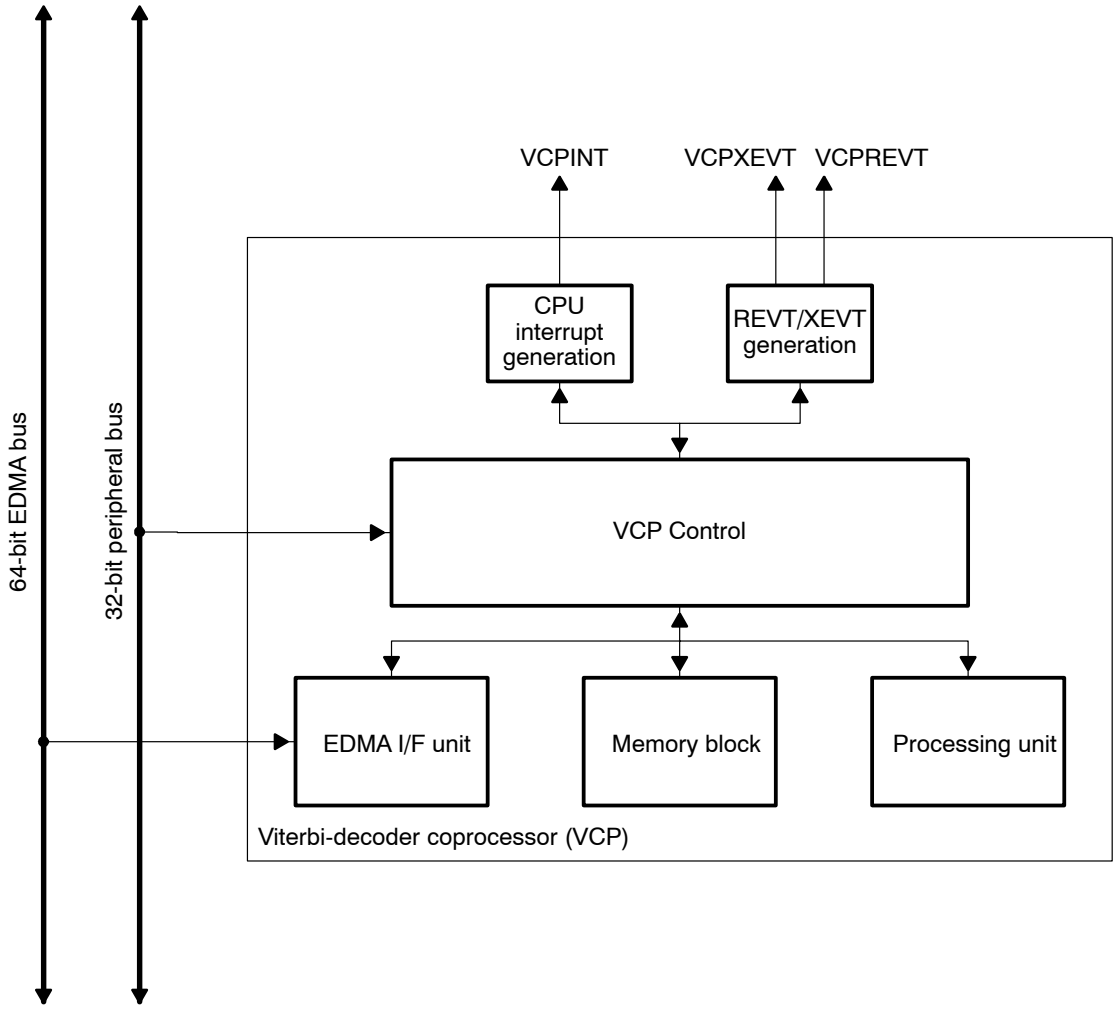
Note: K = 3, R = k/n = 1/3, $G_0 = (100)_8$, $G_1 = (101)_8$, $G_2 = (111)_8$
0/000 means input is 0, output1 is 0, output2 is 0, output3 is 0.
There are $2^{(K-1)}$ states and 2^k incoming branches per state.

3 Overview

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

The VCP input data corresponds to the branch metrics and the output data to the hard decisions or soft decisions.

Figure 3. VCP Block Diagram



4 Input Data

The branch metrics (BM) are calculated by the DSP and stored in the DSP memory subsystem as 7-bit signed values. Per symbol interval T, for a rate $R = k/n$ and a constraint length K, there are a total of 2^{K-1+k} branches in the trellis. For rate 1/n codes, only 2^{n-1} branch metrics need to be computed per symbol period and passed to the VCP. Moreover, n symbols are required to calculate 1 branch metric.

Assuming BSPK modulated bits ($0 \rightarrow 1$, $1 \rightarrow -1$), the branch metrics are calculated as follows:

- Rate 1/2: there are 2 branch metrics per symbol period
  - $BM_0(t) = r_0(t) + r_1(t)$
  - $BM_1(t) = r_0(t) - r_1(t)$

  where $r(t)$ is the received codeword at time t (2 symbols, $r_0(t)$ is the symbol corresponding to the encoder upper branch – see Figure 1).
- Rate 1/3: there are 4 branch metrics per symbol period
  - $BM_0(t) = r_0(t) + r_1(t) + r_2(t)$
  - $BM_1(t) = r_0(t) + r_1(t) - r_2(t)$
  - $BM_2(t) = r_0(t) - r_1(t) + r_2(t)$
  - $BM_3(t) = r_0(t) - r_1(t) - r_2(t)$

  where $r(t)$ is the received codeword (3 symbols, $r_0(t)$ is the symbol corresponding to the encoder upper branch – see Figure 1).
- Rate 1/4: there are 8 branch metrics per symbol period
  - $BM_0(t) = r_0(t) + r_1(t) + r_2(t) + r_3(t)$
  - $BM_1(t) = r_0(t) + r_1(t) + r_2(t) - r_3(t)$
  - $BM_2(t) = r_0(t) + r_1(t) - r_2(t) + r_3(t)$
  - $BM_3(t) = r_0(t) + r_1(t) - r_2(t) - r_3(t)$
  - $BM_4(t) = r_0(t) - r_1(t) + r_2(t) + r_3(t)$
  - $BM_5(t) = r_0(t) - r_1(t) + r_2(t) - r_3(t)$
  - $BM_6(t) = r_0(t) - r_1(t) - r_2(t) + r_3(t)$
  - $BM_7(t) = r_0(t) - r_1(t) - r_2(t) - r_3(t)$

  where $r(t)$ is the received codeword (4 symbols, $r_0(t)$ is the symbol corresponding to the encoder upper branch – see Figure 1).

The data must be sent to the VCP as described in Table 1, Table 2, and Table 3 for rates 1/2, 1/3, and 1/4, respectively (the base address must be double-word aligned).

The branch metrics can be saved in the DSP memory subsystem in either their native format or packed in words (user implementation). When working in big-endian mode, the VCP endian mode register (VCPEND) has to be programmed accordingly (see section 6.10).

Table 1. Branch Metrics for Rate 1/2

	Data			
Address (hex)	**MSB**			**LSB**
Base	$BM_1(t=T)$	$BM_0(t=T)$	$BM_1(t=0)$	$BM_0(t=0)$
Base + 4h	$BM_1(t=3T)$	$BM_0(t=3T)$	$BM_1(t=2T)$	$BM_0(t=2T)$
Base + 8h	…			

Table 2. Branch Metrics for Rate 1/3

	Data			
Address (hex)	**MSB**			**LSB**
Base	$BM_3(t=0)$	$BM_2(t=0)$	$BM_1(t=0)$	$BM_0(t=0)$
Base + 4h	$BM_3(t=T)$	$BM_2(t=T)$	$BM_1(t=T)$	$BM_0(t=T)$
Base + 8h	…			

Table 3. Branch Metrics for Rate 1/4

	Data			
Address (hex)	**MSB**			**LSB**
Base	$BM_3(t=0)$	$BM_2(t=0)$	$BM_1(t=0)$	$BM_0(t=0)$
Base + 4h	$BM_7(t=0)$	$BM_6(t=0)$	$BM_5(t=0)$	$BM_4(t=0)$
Base + 8h	$BM_3(t=T)$	$BM_2(t=T)$	$BM_1(t=T)$	$BM_0(t=T)$
Base + Ch	$BM_7(t=T)$	$BM_6(t=T)$	$BM_5(t=T)$	$BM_4(t=T)$
Base + 10h	…			

5 Output Data

The VCP can be configured to send either hard decisions (a bit) or soft decisions (a 16-bit value, 12-bit sign-extended) to the DSP after the decoding.

Decisions ordering at the VCP output depend on the programmed traceback mode and the VCPEND in case the DSP is set to work in big-endian mode (see the VCP endian mode register, section 6.10).

The decisions buffer start address must be double-word aligned and the buffer size must contain an even number of 32-bit words.

6 Registers

The VCP contains several memory-mapped registers accessible by way of the CPU load and store instructions, the QDMA, and the EDMA. A peripheral-bus access is faster than an EDMA-bus access for isolated accesses (typically when accessing control registers). EDMA-bus accesses are intended to be used for EDMA transfers and are meant to provide maximum throughput to/from the VCP.

The memory map is listed in Table 4. The branch metric and decision memories contents are not accessible and the memories can be regarded as FIFOs by the DSP, meaning you do not have to perform any indexing on the addresses.

Table 4. VCP Registers

Start Address (hex)				
EDMA bus	**Peripheral Bus**	**Acronym**	**Register Name**	**Section**
5000 0000	01B8 0000	VCPIC0	VCP input configuration register 0	6.1
5000 0004	01B8 0004	VCPIC1	VCP input configuration register 1	6.2
5000 0008	01B8 0008	VCPIC2	VCP input configuration register 2	6.3
5000 000C	01B8 000C	VCPIC3	VCP input configuration register 3	6.4
5000 0010	01B8 0010	VCPIC4	VCP input configuration register 4	6.5
5000 0014	01B8 0014	VCPIC5	VCP input configuration register 5	6.6
5000 0048	01B8 0048	VCPOUT0	VCP output register 0	6.7
5000 004C	01B8 004C	VCPOUT1	VCP output register 1	6.8
5000 0080	–	VCPWBM	VCP branch metrics write register	–
5000 0088	–	VCPRDECS	VCP decisions read register	–
–	01B8 0018	VCPEXE	VCP execution register	6.9
–	01B8 0020	VCPEND	VCP endian mode register	6.10
–	01B8 0040	VCPSTAT0	VCP status register 0	6.11
–	01B8 0044	VCPSTAT1	VCP status register 1	6.12
–	01B8 0050	VCPERR	VCP error register	6.13

6.1 VCP Input Configuration Register 0 (VCPIC0)

The VCP input configuration register 0 (VCPIC0) is shown in Figure 4 and described in Table 5.

Figure 4. VCP Input Configuration Register 0 (VCPIC0)

31 24	23 16	15 8	7 0
POLY3	POLY2	POLY1	POLY0
R/W-0	R/W-0	R/W-0	R/W-0

Legend: R/W = Read/write; -*n* = value after reset

Table 5. VCP Input Configuration Register 0 (VCPIC0) Field Descriptions

Bit	*field*†	*symval*†	Value	Description‡
31–24	POLY3	OF(*value*)	0–FFh	Polynomial generator G_3 .See section 9.2.
23–16	POLY2	OF(*value*)	0–FFh	Polynomial generator G_2 .See section 9.2.
15–8	POLY1	OF(*value*)	0–FFh	Polynomial generator G_1 .See section 9.2.
7–0	POLY0	OF(*value*)	0–FFh	Polynomial generator G_0 .See section 9.2.

† For CSL implementation, use the notation VCP_IC0_POLY*n*_*symval*

‡ The polynomial generators are 9-bit values defined as $G(z) = b_8z^{-8} + b_7z^{-7} + b_6z^{-6} + b_5z^{-5} + b_4z^{-4} + b_3z^{-3} + b_2z^{-2} + b_1z^{-1} + b_0$, but only 8 bits are passed in the POLY*n* bitfields so that b_1 is the most significant bit and b_8 the least significant bit (b_0 is not passed but set to 1 by the internal VCP hardware).

6.2 VCP Input Configuration Register 1 (VCPIC1)

The VCP input configuration register 1 (VCPIC1) is shown in Figure 5 and described in Table 6.

Figure 5. VCP Input Configuration Register 1 (VCPIC1)

Legend: R/W = Read/write; -*n* = value after reset

Table 6. VCP Input Configuration Register 1 (VCPIC1) Field Descriptions

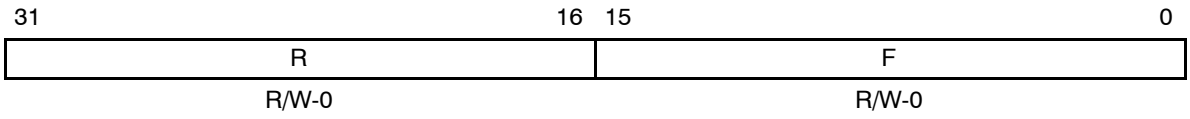
Bit	*field*†	*symval*†	Value	Description
31–29	Reserved	–	0	Reserved. The reserved bit location is always read as 0. A value written to this field has no effect.
28	YAMEN			Yamamoto algorithm enable bit. See section 8.2.
		DISABLE	0	Yamamoto algorithm is disabled.
		ENABLE	1	Yamamoto algorithm is enabled.
27–16	YAMT	OF(*value*)	0–FFFh	Yamamoto threshold value bits. See section 8.2.
15–0	Reserved	–	0	Reserved. These reserved bit locations must be 0. A value written to this field has no effect.

† For CSL implementation, use the notation VCP_IC1_*field_symval*

6.3 VCP Input Configuration Register 2 (VCPIC2)

The VCP input configuration register 2 (VCPIC2) is shown in Figure 6 and described in Table 7.

Figure 6. VCP Input Configuration Register 2 (VCPIC2)



Legend: R/W = Read/write; -*n* = value after reset

Table 7. VCP Input Configuration Register 2 (VCPIC2) Field Descriptions

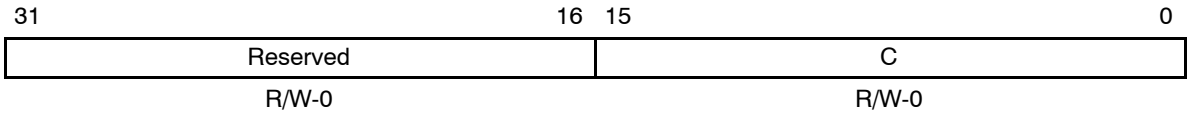
Bit	*field*†	*symval*†	Value	Description
31–16	R	OF(*value*)	0–FFFFh	Reliability length bits. See section 8.1.
15–0	F	OF(*value*)	0–FFFFh	Frame length bits. See section 8.1.

† For CSL implementation, use the notation VCP_IC2_*field_symval*

6.4 VCP Input Configuration Register 3 (VCPIC3)

The VCP input configuration register 3 (VCPIC3) is shown in Figure 7 and described in Table 8.

Figure 7. VCP Input Configuration Register 3 (VCPIC3)



Legend: R/W = Read/write; -*n* = value after reset

Table 8. VCP Input Configuration Register 3 (VCPIC3) Field Descriptions

Bit	Field	*symval*†	Value	Description
31–16	Reserved	–	0	Reserved. The reserved bit location is always read as 0. A value written to this field has no effect.
15–0	C	OF(*value*)	0–FFFFh	Convergence distance bits. See section 8.1.

† For CSL implementation, use the notation VCP_IC3_C_*symval*

6.5 VCP Input Configuration Register 4 (VCPIC4)

The VCP input configuration register 4 (VCPIC4) is shown in Figure 8 and described in Table 9.

Figure 8. VCP Input Configuration Register 4 (VCPIC4)

31	28	27	16
Reserved		IMINS	
R/W-0		R/W-0	

15	12	11	0
Reserved		IMAXS	
R/W-0		R/W-0	

Legend: R/W = Read/write; -*n* = value after reset

Table 9. VCP Input Configuration Register 4 (VCPIC4) Field Descriptions

Bit	*field*†	*symval*†	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.
27–16	IMINS	OF(*value*)	0–FFFh	Minimum initial state metric value bits. See section 9.2.
15–12	Reserved	–	0	Reserved. The reserved bit location is always read as 0. A value written to this field has no effect.
11–0	IMAXS	OF(*value*)	0–FFFh	Maximum initial state metric value bits. See section 9.2.

† For CSL implementation, use the notation VCP_IC4_*field_symval*

6.6 VCP Input Configuration Register 5 (VCPIC5)

The VCP input configuration register 5 (VCPIC5) is shown in Figure 9 and described in Table 10.

Figure 9. VCP Input Configuration Register 5 (VCPIC5)

31	30	29–26	25–24	23–20	19–16
SDHD	OUTF	Reserved	TB	SYMR	SYMX
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

15–8	7–0
Reserved	IMAXI
R/W-0	R/W-0

Legend: R/W = Read/write; -*n* = value after reset

Table 10. VCP Input Configuration Register 5 (VCPIC5) Field Descriptions

Bit	*field*†	*symval*†	Value	Description
31	SDHD			Output decision type select bit.
		HARD	0	Hard decisions.
		SOFT	1	Soft decisions.
30	OUTF			Output parameters read flag bit.
		NO	0	VCPREVT is not generated by VCP for output parameters read.
		YES	1	VCPREVT generated by VCP for output parameters read.
29–26	Reserved	–	0	Reserved. The reserved bit location is always read as 0. A value written to this field has no effect.
25–24	TB		0–3h	Traceback mode select bits.
		NO	0	Not allowed.
		TAIL	1h	Tailed.
		CONV	2h	Convergent.
		MIX	3h	Mixed.
23–20	SYMR	OF(*value*)	0–Fh	Determines decision buffer length in output FIFO. When programming register values for the SYMR bits, always subtract 1 from the value calculated. Valid values for the SYMR bits are from 0 to Fh. See section 8.4.

† For CSL implementation, use the notation VCP_IC5_*field_symval*

Table 10. VCP Input Configuration Register 5 (VCPIC5) Field Descriptions (Continued)

Bit	*field*†	*symval*†	Value	Description
19–16	SYMX	OF(*value*)	0–Fh	Determines branch metrics buffer length in input FIFO. When programming register values for the SYMX bits, always subtract 1 from the value calculated. Valid values for the SYMX bits are from 0 to Fh. See section 8.3.
15–8	Reserved	–	0	Reserved. The reserved bit location is always read as 0. A value written to this field has no effect.
7–0	IMAXI	OF(*value*)	0–FFh	Maximum initial state metric value bits. IMAXI bits determine which state should be initialized with the maximum state metrics value (IMAXS) bits in VCPIC4; all the other states will be initialized with the value in the IMINS bits.

† For CSL implementation, use the notation VCP_IC5_*field_symval*

6.7 VCP Output Register 0 (VCPOUT0)

The VCP output register 0 (VCPOUT0) is shown in Figure 10 and described in Table 11.

Figure 10. VCP Output Register 0 (VCPOUT0)

31	28	27	16
Reserved		FMINS	
R/W-0		R-0	

15	12	11	0
Reserved		FMAXS	
R/W-0		R-0	

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

Table 11. VCP Output Register 0 (VCPOUT0) Field Descriptions

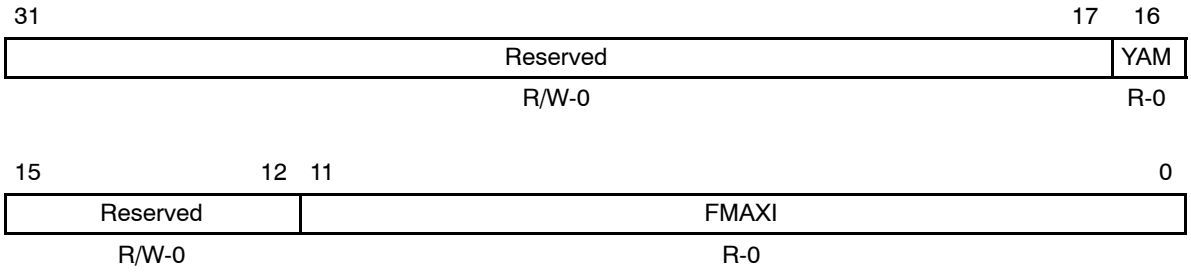
Bit	*field*†	*symval*†	Value	Description
31–28	Reserved	–	0	Reserved. The reserved bit location is always read as 0. A value written to this field has no effect.
27–16	FMINS	OF(*value*)	0–FFFh	Final minimum state metric value bits.
15–12	Reserved	–	0	Reserved. The reserved bit location is always read as 0. A value written to this field has no effect.
11–0	FMAXS	OF(*value*)	0–FFFh	Final maximum state metric value bits.

† For CSL implementation, use the notation VCP_OUT0_*field_symval*

6.8 VCP Output Register 1 (VCPOUT1)

The VCP output register 1 (VCPOUT1) is shown in Figure 11 and described in Table 12.

Figure 11. VCP Output Register 1 (VCPOUT1)



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

Table 12. VCP Output Register 1 (VCPOUT1) Field Descriptions

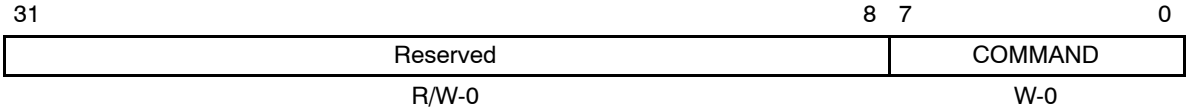
Bit	*field*†	*symval*†	Value	Description
31–17	Reserved	–	0	Reserved. The reserved bit location is always read as 0. A value written to this field has no effect.
16	YAM			Yamamoto bit result. See section 8.2.
		NO	0	
		YES	1	
15–12	Reserved	–	0	Reserved. The reserved bit location is always read as 0. A value written to this field has no effect.
11–0	FMAXI	OF(*value*)	0–FFFh	State index for the state with the final maximum state metric.

† For CSL implementation, use the notation VCP_OUT1_*field_symval*

6.9 VCP Execution Register (VCPEXE)

The VCP execution register (VCPEXE) is shown in Figure 12 and described in Table 13.

Figure 12. VCP Execution Register (VCPEXE)



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

Table 13. VCP Execution Register (VCPEXE) Field Descriptions

Bit	Field	*symval*†	Value	Description
31–8	Reserved	–	0	Reserved. The reserved bit location is always read as 0. A value written to this field has no effect.
7–0	COMMAND		0–FFh	VCP command select bits. See section 12.
		–	0	Reserved.
		START	1h	Start.
		PAUSE	2h	Pause.
		–	3h	Reserved
		UNPAUSE	4h	Unpause.
		STOP	5h	Stop
		–	6h–FFh	Reserved.

† For CSL implementation, use the notation VCP_EXE_COMMAND_*symval*

6.10 VCP Endian Mode Register (VCPEND)

The VCP endian mode register (VCPEND) is shown in Figure 13 and described in Table 14. VCPEND has an effect only in big-endian mode.

Figure 13. VCP Endian Mode Register (VCPEND)

31	2	1	0
Reserved		SD	BM
R/W-0		R/W-0	R/W-0

Legend: R/W = Read/write; -*n* = value after reset

Table 14. VCP Endian Mode Register (VCPEND) Field Descriptions

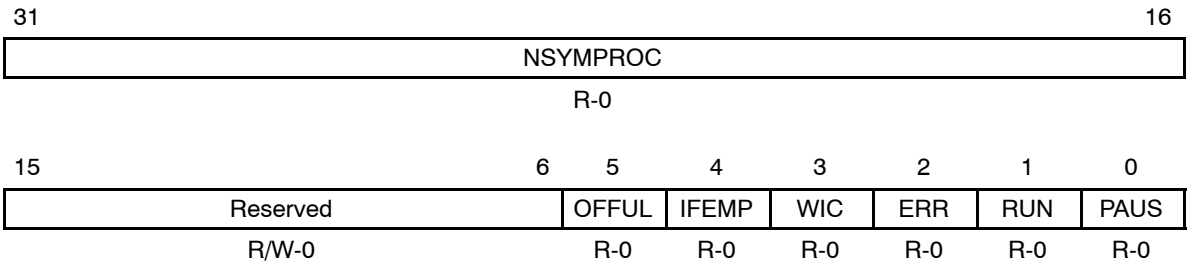
Bit	*field*†	*symval*†	Value	Description
31–2	Reserved	–	0	Reserved. The reserved bit location is always read as 0. A value written to this field has no effect.
1	SD			Soft-decisions memory format select bit. See section 7.2.
		32BIT	0	32-bit-word packed.
		NATIVE	1	Native format (16 bits).
0	BM			Branch metrics memory format select bit. See section 7.1.
		32BIT	0	32-bit-word packed.
		NATIVE	1	Native format (8 bits).

† For CSL implementation, use the notation VCP_END_*field_symval*

6.11 VCP Status Register 0 (VCPSTAT0)

The VCP status register 0 (VCPSTAT0) is shown in Figure 14 and described in Table 15.

Figure 14. VCP Status Register 0 (VCPSTAT0)



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

Table 15. VCP Status Register 0 (VCPSTAT0) Field Descriptions

Bit	*field*†	*symval*†	Value	Description
31–16	NSYMPROC	OF(*value*)	0–FFFFh	Number of symbols processed bits. The NSYMPROC bits indicate how many symbols have been processed in the state metric unit.
15–6	Reserved	–	0	Reserved. The reserved bit location is always read as 0. A value written to this field has no effect.
5	OFFUL			Output FIFO buffer full status bit.
		NO	0	Output FIFO buffer is not full.
		YES	1	Output FIFO buffer is full.
4	IFEMP			Input FIFO buffer empty status bit.
		NO	0	Input FIFO buffer is not empty.
		YES	1	Input FIFO buffer is empty.
3	WIC			Waiting for input configuration bit. The WIC bit indicates that the VCP is waiting for new input control parameters to be written. This bit is always set after decoding of a user channel.
		NO	0	Not waiting for input configuration words.
		YES	1	Waiting for input configuration words.

† For CSL implementation, use the notation VCP_STAT0_*field_symval*

Table 15. VCP Status Register 0 (VCPSTAT0) Field Descriptions (Continued)

Bit	*field*†	*symval*†	Value	Description
2	ERR			VCP error status bit. The ERR bit is cleared as soon as the DSP reads the VCP error register (VCPERR).
		NO	0	No error.
		YES	1	VCP paused due to error.
1	RUN			VCP running status bit.
		NO	0	VCP is not running.
		YES	1	VCP is running.
0	PAUS			VCP pause status bit.
		NO	0	VCP is not paused. The UNPAUSE command is acknowledged by clearing the PAUS bit.
		YES	1	VCP is paused. The PAUSE command is acknowledged by setting the PAUS bit. The PAUS bit can also be set, if the input FIFO buffer is becoming empty or if the output FIFO buffer is full.

† For CSL implementation, use the notation VCP_STAT0_*field_symval*

6.12 VCP Status Register 1 (VCPSTAT1)

The VCP status register 1 (VCPSTAT1) is shown in Figure 15 and described in Table 16.

Figure 15. VCP Status Register 1 (VCPSTAT1)

31	16	15	0
NSYMOF		NSYMIF	
R-0		R-0	

Legend: R = Read only; -*n* = value after reset

Table 16. VCP Status Register 1 (VCPSTAT1) Field Descriptions

Bit	*field*†	*symval*†	Value	Description
31–16	NSYMOF	OF(*value*)	0–FFFFh	Number of symbols in the output FIFO buffer.
15–0	NSYMIF	OF(*value*)	0–FFFFh	Number of symbols in the input FIFO buffer.

† For CSL implementation, use the notation VCP_STAT1_*field_symval*

6.13 VCP Error Register (VCPERR)

The VCP error register (VCPERR) is shown in Figure 16 and described in Table 17.

Figure 16. VCP Error Register (VCPERR)

31	3	2	0
Reserved		ERROR	
R/W-0		R-0	

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

Table 17. VCP Error Register (VCPERR) Field Descriptions

Bit	Field	*symval*†	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	ERROR		0–7h	VCP error indicator bits.
		NO	0	No error is detected.
		TBNA	1h	Traceback mode is not allowed.
		FTL	2h	F too large for tailed traceback mode.
		FCTL	3h	*R* + *C* too large for mixed or convergent traceback modes.
		–	4h–7h	Reserved

† For CSL implementation, use the notation VCP_ERR_ERROR_*symval*

7 Endianness Issues

The VCP endian mode register (VCPEND) is intended to solve possible big-endian issues and is therefore used only when the DSP is in big-endian mode. Depending on whether the data is saved in the DSP memory subsystem in its native format or is 32-bit word packed, interpretation of the data will be different as the DSP is sending 32-bit words to the VCP.

7.1 Branch Metrics

When the data are saved in their 7-bit native format (BM = 1), they must be organized in the DSP memory as described in Table 18. When the data are packed on 32-bit words (BM = 0), they must be organized in the DSP memory as described in Table 19.

Table 18. Branch Metrics in DSP Memory (BM = 1)

Address (hex bytes)	Data
Base	BM0
Base + 1	BM1
Base + 2	BM2
Base + 3	BM3
Base + 4	BM4
Base + 5	BM5
Base + 6	BM6
Base + 7	BM7

Table 19. Branch Metrics in DSP Memory (BM = 0)

Address (hex bytes)	Data
Base	BM3
Base + 1	BM2
Base + 2	BM1
Base + 3	BM0
Base + 4	BM7
Base + 5	BM6
Base + 6	BM5
Base + 7	BM4

7.2 Soft Decisions

When the data are saved in their 16-bit native format (SD = 1), they must be organized in the DSP memory as described in Table 20. When the data are packed on 32-bit words (SD = 0), they must be organized in the DSP memory as described in Table 21.

Table 20. Soft Decisions in DSP Memory (SD = 1)

Address (hex bytes)	Data
Base	SD0
Base + 2	SD1
Base + 4	SD2
Base + 6	SD3

Table 21. Soft Decisions in DSP Memory (SD = 0)

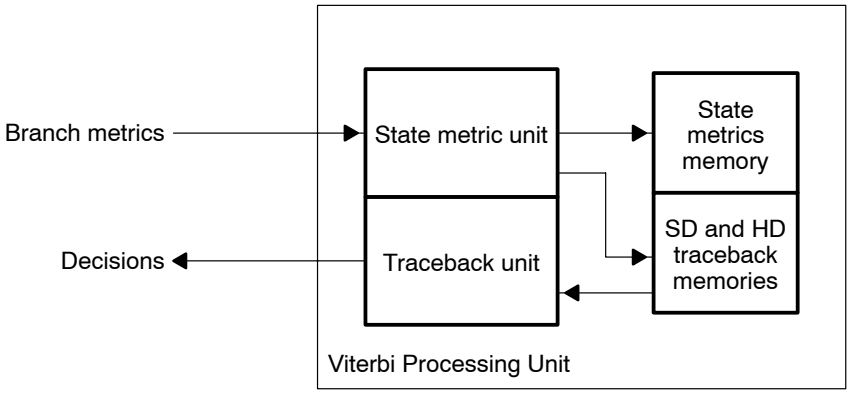
Address (hex bytes)	Data
Base	SD1
Base + 2	SD0
Base + 4	SD3
Base + 6	SD2

8 Architecture

The VCP processing unit is shown in Figure 17.

The state metrics unit performs the Viterbi forward recursion using branch metrics as inputs and updates the states metrics for all states (Add/Compare/Select or ACS operations) at every trellis stage. The state metrics memory is not accessible by the DSP. The traceback unit performs the Viterbi backward recursion and generates hard-decisions or soft-decisions. The traceback memories are not directly accessible by the DSP.

Figure 17. Processing Unit



8.1 Sliding Windows Processing

The hard-decision memory can store up to 32768 traceback bits and there are 2^{K-1} bits stored at each trellis stage. Therefore, the hard-decision memory can store decisions of $32768/2^{K-1}$ symbols.

The soft-decision memory can store up to 8192 traceback soft values and, therefore, contain up to 8192 soft decisions of $8192/2^{K-1}$ symbols.

Assume a terminated frame of length *F* (excluding tail bits) and a constraint length *K*, *F* and *K* determine whether all decisions can be stored in the traceback memories. If all decisions do not fit, then the traceback mode is set to mixed and the original frame is segmented into sliding windows (SW); otherwise, the traceback mode is set to tailed and no segmentation is required.

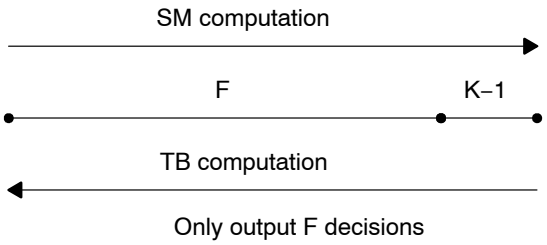
In case of a nonterminated frame or if you want to start decoding without waiting for the end of the frame, the traceback mode should be set to convergent and the frame might have to be segmented into sliding windows depending on whether the decisions will fit in the traceback memories.

8.1.1 Tailed Traceback Mode

This mode is used when a full frame can reside within the coprocessor traceback memory (see Figure 18).

The state metrics are computed over $F + K - 1$ symbols, the traceback is initialized with the tails state and executed over $F + K - 1$ symbols. It should be noted that only F decisions are output. They are output in reverse order and in blocks of user-defined size (see section 8.4).

Figure 18. Tailed Traceback Mode



8.1.2 Mixed Traceback Mode

This mode is used when the the full frame does not fit into the coprocessor traceback memory and the frame is terminated. The frame is split into sliding windows (see Figure 19).

The state metrics are computed over $F + K - 1$ symbols, the traceback is initialized with the tails state and executed over $F + K - 1$ symbols. It should be noted that only F decisions are output in blocks of user-defined size (see section 8.3).

The state metrics computation of sliding window $i + 1$ is done in parallel with the traceback computation of sliding window i.

Tailed traceback type is used on the last sliding window.

8.1.3 Convergent Traceback Mode

This mode is used with nonterminated frames or when you want to decode a portion of the frame.

When the frame does not fit into the coprocessor traceback memory, then the frame is split into sliding windows (see Figure 20).

The state metrics are computed over $F + C$ symbols, the traceback is initialized with the tails state and executed over $F + C$ symbols. It should be noted that only F decisions are output in blocks of user-defined size (see section 8.4).

The state metrics computation of sliding window $i + 1$ is done in parallel with the traceback computation of sliding window i.

Figure 19. Mixed Traceback Mode—Example With Five Sliding Windows

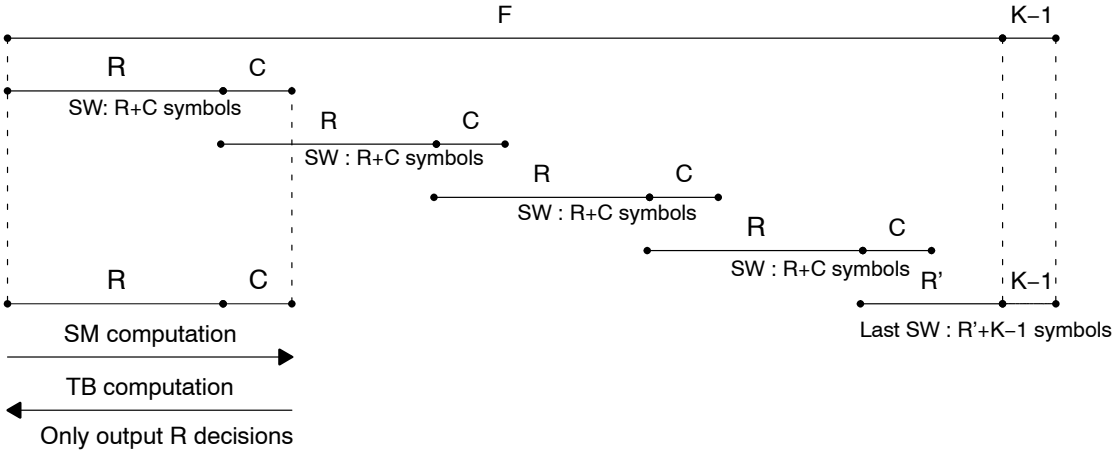
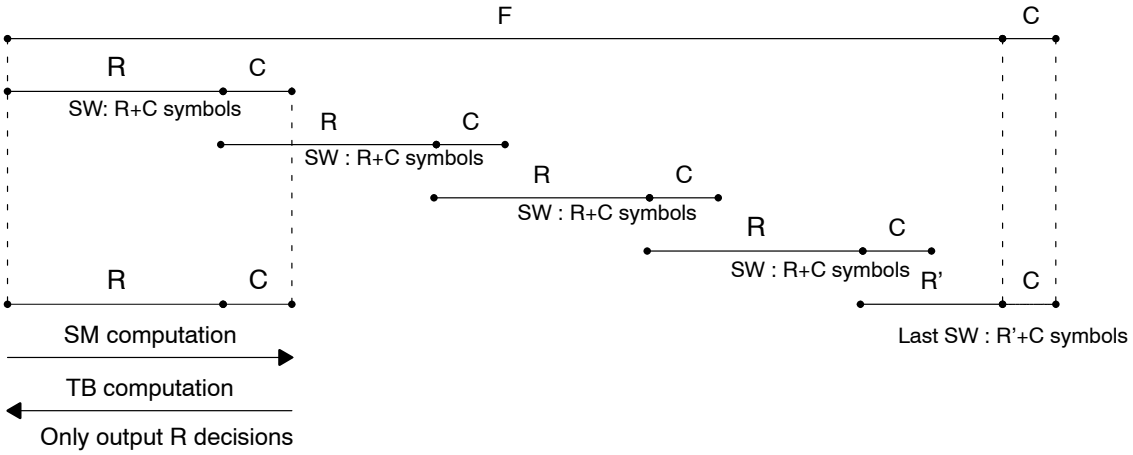


Figure 20. Convergent Traceback Mode—Example With Five Sliding Windows



8.1.4 F, R, and C Limitations

Given a frame of length *F* (length prior to convolutional encoding – no tail bit information accounted), there are some limitations on the *R* and *C* values that you must follow. Unpredictable behavior will occur if those constraints are not observed. These limitations are summarized in Table 22.

Table 22. F, R, and C Limitations

	Traceback Mode					
	Hard Decisions			Soft Decisions		
	Tailed	Mixed/Convergent†		Tailed	Mixed/Convergent†	
K	Fmax	R + C	Possible C values	Fmax	R, C = 3 (K – 1) (non-punctured code)	R, C = 6 (K – 1) (punctured code)
9	120	124	3,6,9,12,15 × (K – 1)	24	R=4, C=24	not allowed
8	217	217	3,6,9,12,15,18 × (K – 1)	49	R=28, C=21	R=7, C=42
7	378	372	3,6,9,12,15,18 × (K – 1)	90	R=60, C=18	R=54, C=36
6	635	605	3,6,9,12,15,18 × (K – 1)	155	R=60, C=15	R=60, C=30
5	2044	1020	3,6,9,12,15,18 × (K – 1)	508	R=60, C=12	R=60, C=24

† Mixed mode is not allowed for frame sizes that can be handled in tailed mode

Additional configurations that are valid for F, R, and C in hard decisions, convergent mode are: R = 192, C = 96, Rate = 1/3, K = 7, and Frame lengths of 278, 310, 342, 358, 480, 482, 486, 624, 626, 768, 770, and 802.

8.2 Yamamoto Parameters

During the standard forward recursion, an entity called the Yamamoto bit is computed for each state and updated every symbol interval.

The Yamamoto bit was proposed by Hirosuke Yamamoto (Hirosuke Yamamoto, “Viterbi Decoding Algorithm for Convolutional Codes with Repeat Request,” IEEE Transactions on Information Theory, Vol. IT-26, No. 5, September 1980).

The basic concept is that a bit (the Yamamoto bit) is associated with each state in the decoding process. Initially, all the Yamamoto bits are set (1). During the decoding process, the Yamamoto bit for a particular state comes from a couple of decisions made on the path metrics and the Yamamoto bit of previous states. The metrics of all paths leading to a particular state are compared. If the difference between any two metrics is less than a given threshold (YAMT bits in VCPIC1), then the Yamamoto bit is cleared to zero (0); otherwise, the Yamamoto bit is inherited from the previous state of the path with the largest metric. The end result of this process (YAM bit in VCPOUT1) yields a zero (0) if anywhere along the decoding path there was a point where the decision between two paths was ambiguous. The YAM bit can therefore be used as a binary frame quality indicator.

The Yamamoto algorithm can be enabled or disabled by toggling the YAMEN bit in VCPIC1.

The Yamamoto bit may be falsely cleared to 0 when the number of symbols to be processed is not a multiple of 4 when (FL + (K – 1)%4 = 1, 2, 3 for the last set of symbols to be processed is 1, 2, or 3. The extra (3, 2, 1) symbol stages of the Branch Metrics are automatically being inserted by the VCP as 0s. Automatically inserting the zeroed BM stages can cause the Yamamoto Threshold to be falsely exceeded, thus falsely clearing the Yamamoto bit to 0. You should always choose frame length such that FL + (K – 1)%4 = 0, when using a Yamamoto bit to avoid this issue.

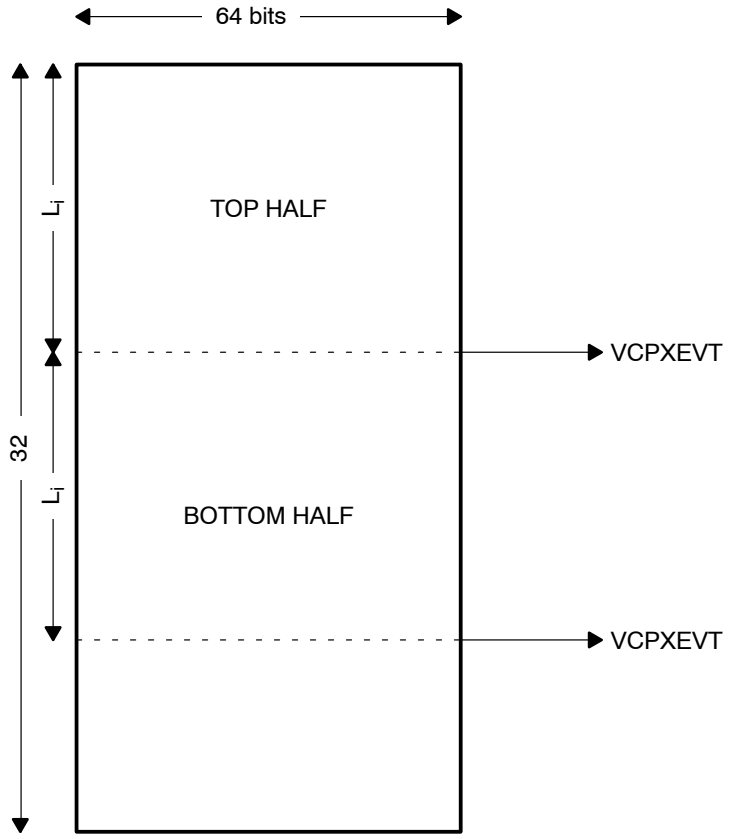
8.3 Input FIFO (Branch Metrics)

The FIFO is used in a double-buffering scheme as described in Figure 21. The VCP generates a VCPXEVT synchronization event each time the top half or bottom half of the buffer is empty.

The SYMX bits are in VCPIC5 and define the buffer length as well as the VCPXEVT event rate. The maximum size for the buffer is 32 64-bit words.

From a user perspective, SYMX corresponds to the number of symbols to be transferred per synchronization event: $4 \times SYMX$ symbols.

Figure 21. Input FIFO (Branch Metrics)



L_i = 8 (SYMX) bytes for rate 1/2 — $SYMX_{max}$ = 16 for rate 1/2
L_i = 16 (SYMX) bytes for rate 1/3 — $SYMX_{max}$ = 8 for rate 1/3
L_i = 32 (SYMX) bytes for rate 1/4 — $SYMX_{max}$ = 4 for rate 1/4

8.4 Output FIFO (Decisions)

The FIFO is used in a double-buffering scheme as described in Figure 22. The VCP generates a VCPREVT synchronization event each time the top half or bottom half of the buffer is full.

The SYMR bits are in VCPIC5 and define the buffer length as well as the VCPREVT event rate. The maximum size for buffer is 32 64-bit words.

From as user perspective, SYMR corresponds to the number of symbols to be transferred per synchronization event:

- For hard-decisions, 64 × *SYMR* symbols
- For soft-decisions, 4 × *SYMX* symbols

Table 23 lists several restrictions to the choice of SYMR imposed by the VCP hardware.

Figure 22. Output FIFO (Decisions Data)

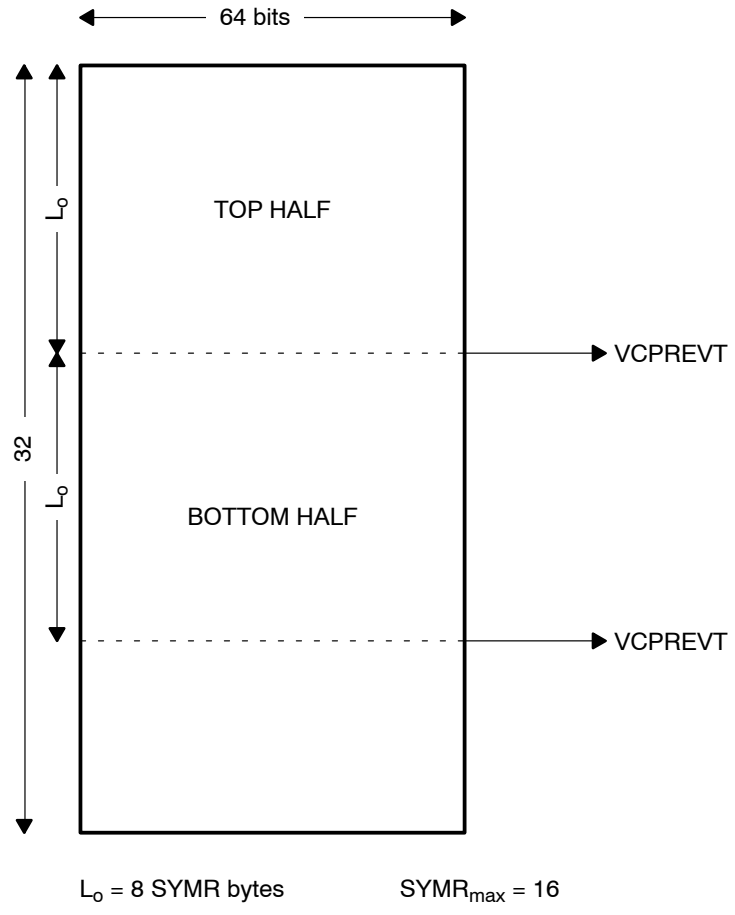


Table 23. Restrictions for SYMR Value

Decisions	Constraint length	Traceback mode	Convergence distance (C)	Frame length (F)	SYMR
Soft	7	Mixed/Convergent	18	Any†	16
Soft	7	Mixed	Any	1324 < F < 1331	11
Soft	7	Mixed	Any	F = 1109 F = 1110 1216 < F < 1223 1260 < F < 1265 1272 < F < 1277 1320 < F < 1325 1362 < F < 1367 1374 < F < 1385 1428 < F < 1439,	12
				otherwise†	13
Soft	7	Convergent	Any	Any†	13
Soft	5	Mixed	Any	Any†	16
Soft	6	Mixed	Any	Any†	16
Soft	6	Convergent	Any	Any†	<16
Hard	5	Mixed/Convergent	Any	Any†	15

† within the limitations imposed by Table 22.

9 Programming

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

- 3 to 4 EDMA parameters (see Table 24)
- The input configurations parameters

Several user channels can be programmed prior to starting the VCP. A suggested implementation is to use the EDMA interrupt generation capabilities (see the *TMS320C6000 DSP Enhanced Direct Memory Access (EDMA) Controller Reference Guide*, SPRU234) and program the EDMA to generate an interrupt after the user channel's last VCPREVT synchronized EDMA transfer has completed.

Table 24. Required EDMA Links Per User Channel

Direction†	Data	Usage	Required/Optional
Transmit	Input configuration parameters	Send the input configuration parameters	Required
Transmit	Branch metrics	Send branch metrics	Required
Receive	Decisions	Read decisions	Required
Receive	Output parameters	Read output parameters	Optional (OUTF bit)

† Transmit direction (DSP−>VCP), receive direction (VCP−>DSP)

9.1 EDMA Resources

9.1.1 VCP Dedicated EDMA Resources

Within the available 64 EDMA channel event sources, two are assigned to the VCP: event 28 and event 29.

- Event 28 is associated to the VCP receive event (VCPREVT) and is used as the synchronization event for EDMA transfers from the VCP to the DSP (receive). EDMA channel 28 is primarily intended to serve VCP to DSP transfers.
- Event 29 is associated to the VCP transmit event (VCPXEVT) and is used as the synchronization event for EDMA transfers from the DSP to the VCP (transmit). EDMA channel 29 is primarily intended to serve DSP to VCP transfers.

9.1.2 Special VCP EDMA Programming Considerations

The EDMA parameters consists of six words as shown in Figure 23. All EDMA transfers, in the context of the VCP, must be done using 32-bit word elements, must contain an even number of words, and have source and destination addresses double-word aligned.

All EDMA transfers must be double-word aligned and the element count for the VCP EDMA transfer must be a multiple of 2. Single-word transfers that are not double-word aligned cause errors in TCP/VCP memory.

For more information, see the *TMS320C6000 DSP Enhanced Direct Memory Access (EDMA) Controller Reference Guide* (SPRU234).

Figure 23. EDMA Parameters Structure

(a) EDMA Registers

	31 — 0	EDMA parameter
Word 0	EDMA Channel Options Parameter (OPT)	OPT
Word 1	EDMA Channel Source Address (SRC)	SRC
Word 2	Array/frame count (FRMCNT) | Element count (ELECNT)	CNT
Word 3	EDMA Channel Destination Address (DST)	DST
Word 4	Array/frame index (FRMIDX) | Element index (ELEIDX)	IDX
Word 5	Element count reload (ELERLD) | Link address (LINK)	RLD

(b) EDMA Channel Options Parameter (OPT)

31–29	28–27	26	25–24	23	22–21	20	19–16
PRI	ESIZE	2DS	SUM	2DD	DUM	TCINT	TCC

15	14–13	12	11	10–5	4	3	2	1	0
—	TCCM	ATCINT	—	ATCC	—	PDTS	PDTD	LINK	FS

9.1.2.1 Input Configuration Parameters Transfer

This EDMA transfer to the input configuration parameters is a 6-word VCPXEVT frame synchronized transfer. The parameters should be set as:

- OPTIONS:
  - ESIZE = 00 (element size is 32 bits)
  - 2DS = 2DD = 0 (1 dimensional)
  - SUM = 01 (autoincrement)
  - DUM = 01 (autoincrement)
  - LINK = 1 (linking of event parameters is enabled)
  - FS = 1 (channel is frame synchronized)
- SOURCE ADDRESS: user input configuration parameters start address
- ELEMENT COUNT: 0006h
- LINE/FRAME COUNT: 0000h
- DESTINATION ADDRESS: VCPIC0 (5000 0000h)
- ELEMENT INDEX: don't care
- LINE/FRAME INDEX: don't care
- LINK ADDRESS: address in the EDMA PaRAM of the EDMA parameters associated with the branch metrics
- ELEMENT COUNT RELOAD: don't care

Upon completion, this EDMA transfer is linked to the EDMA for branch metrics transfer parameters.

9.1.2.2 Branch Metrics Transfer

This EDMA transfer to the branch metrics FIFO is a VCPXEVT frame synchronized transfer. The parameters should be set as:

- OPTIONS:
  - ESIZE = 00 (element size is 32 bits)
  - 2DS = 2DD = 0 (1 dimensional)
  - SUM = 01 (autoincrement)
  - DUM = 00 (fixed)
  - LINK = 1 (linking of event parameters is enabled)
  - FS = 1 (channel is frame synchronized)
- SOURCE ADDRESS: user branch metrics start address

- ELEMENT COUNT: $2^{n-1} \times \min(SYMX, F + K - 1)$
- LINE/FRAME COUNT: $ceil\left(\frac{F + K - 1}{4 \times SYMX}\right) - 1$
- DESTINATION ADDRESS: VCPWBM (5000 0080h)
- ELEMENT INDEX: don't care
- LINE/FRAME INDEX: don't care
- LINK ADDRESS: see cases 1 and 2 below
- ELEMENT COUNT RELOAD: don't care

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

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

9.1.2.3 Decisions Transfer

This EDMA transfer to the decisions buffer is a VCPREVT frame synchronized transfer. The programming of this transfer depends on the decision type and the traceback mode.

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

1) The decisions EDMA transfer parameters of the next user-channel, if there is one ready to be decoded and OUTF bit is 0.
2) Null EDMA transfer parameters (with all zeros), if there is no more user-channel ready to be decoded and OUTF bit is 0.
3) The output parameters EDMA transfer parameters, if OUTF bit is 1.

9.1.2.4 Hard-Decisions Mode

- OPTIONS:
  - ESIZE = 00 (element size is 32 bits)
  - 2DS = 2DD = 0 (1 dimensional)
  - SUM = 00 (fixed)
  - DUM = 10 (autodecrement for tailed traceback mode) or 01 (autoincrement for mixed/convergent traceback mode
  - LINK = 1 (linking of event parameters is enabled)
  - FS = 1 (channel is frame synchronized)

- SOURCE ADDRESS: VCPRDECS (5000 0088h)
- ELEMENT COUNT: $2 \times ceil\left(\frac{\min(SYMR \times 64, F)}{64}\right)$
- LINE/FRAME COUNT: $ceil\left(\frac{F}{64 \times SYMR}\right) - 1$
- DESTINATION ADDRESS: user hard-decision buffer start address
- ELEMENT INDEX: don't care
- LINE/FRAME INDEX: don't care
- LINK ADDRESS: see cases 1, 2, and 3 above
- ELEMENT COUNT RELOAD: don't care

9.1.2.5 Soft-Decisions Mode

- OPTIONS:
  - ESIZE = 00 (element size is 32 bits)
  - 2DS = 2DD = 0 (1 dimensional)
  - SUM = 00 (fixed)
  - DUM = 10 (autodecrement for tailed traceback mode) or 01 (autoincrement for mixed/convergent traceback mode
  - LINK = 1 (linking of event parameters is enabled)
  - FS = 1 (channel is frame synchronized)
- SOURCE ADDRESS: VCPRDECS (5000 0088h)
- ELEMENT COUNT: $2 \times ceil\left(\frac{\min(SYMR \times 4, F)}{4}\right)$
- LINE/FRAME COUNT: $ceil\left(\frac{F}{4 \times SYMR}\right) - 1$
- DESTINATION ADDRESS: user hard-decision buffer start address
- ELEMENT INDEX: don't care
- LINE/FRAME INDEX: don't care
- LINK ADDRESS: see cases 1, 2, and 3 above
- ELEMENT COUNT RELOAD: don't care

9.1.2.6 Output Parameters Transfer

This transfer is optional and depends on the OUTF bit. It is a 2−32-bit word VCPREVT frame synchronized transfer.

The parameters should be set as following:

- OPTIONS:
  - ESIZE = 00 (element size is 32 bits)
  - 2DS = 2DD = 0 (1 dimensional)
  - SUM = 01 (autoincrement)
  - DUM = 01 (autoincrement)
  - LINK = 1 (linking of event parameters is enabled)
  - FS = 1 (channel is frame synchronized)
- SOURCE ADDRESS: VCPOUT0 (5000 0048h)
- ELEMENT COUNT: 0002h
- LINE/FRAME COUNT: 0000h
- DESTINATION ADDRESS: user output parameters source address
- ELEMENT INDEX: don’t care
- LINE/FRAME INDEX: don’t care
- LINK ADDRESS: see case 1 and 2 above
- ELEMENT COUNT RELOAD: don’t care

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

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

9.2 Input Configuration Words

The input configuration words should reflect the parameters of the user-channels to be decoded.

The POLY*n* bits in VCPIC0 correspond to the generator polynomials in the encoder (see Figure 1 on page 10). The values in each POLY*n* bitfield must be entered in reverse order. It should be noted that the POLY*n* least-significant bit is set to 1 by the VCP logic. For rate 1/2, POLY0 and POLY1 are required; for rate 1/3, POLY0, POLY1, and POLY2 are required; for rate 1/4, all the POLY*n* bits are required.

The YAMT and YAMEN bits in VCPIC1 are described in section 8.2.

The F and R bits in VCPIC2, the C bit in VCPIC3, and the TB bits in VCPIC5 are described in section 8.1.

The IMAXI bits in VCPIC5 determines which state should be initialized with the maximum state metrics value (IMAXS), all the other states are initialized with the minimum state metrics value (IMINS). The IMAXI can range from 0 to $2^{K-1}-1$. The IMAXS and IMINS are 12-bit signed values.

The SYMX and SYMR bits in VCPIC5 are described in section 8.3 and section 8.4.

The OUTF bit in VCPIC5 indicates whether the VCP should generate a VCPREVT for reading the output parameters. The OUTF bit setting will impact the EDMA programming (see section 9.1.2.3)

10 Output Parameters

The FMAXS and FMINS bits in VCPOUT0 indicate the final maximum and minimum state metric values, respectively. The FMAXI bit in VCPOUT1 indicates the state index for the state with the final maximum state metric.

The YAM bit in VCPOUT1 is described in section 8.2.

11 Event Generation

11.1 VCPXEVT Generation

A VCP transmit event (VCPXEVT) is generated when any of the following conditions appears:

- A START command write in VCPEXE.
- All input control words have been received and are correct.
- One half (BOTTOM HALF or TOP HALF) of the input FIFO buffer (see Figure 21) is empty.
- OUTF bit in VCPIC5 is 0 and the traceback is completed.
- OUTF bit in VCPIC5 is 1 and the all the decisions have been read.

11.2 VCPREVT Generation

A VCP receive event (VCPREVT) is generated when any of the following conditions appears:

- The traceback unit has written one half (BOTTOM HALF or TOP HALF) of the output FIFO buffer (see Figure 22).
- OUTF bit in VCPIC5 is 0 and the traceback is completed (the whole frame has been decoded).
- OUTF bit in VCPIC5 is 1 and all decisions have been read.

12 Operational Modes

12.1 Start

To start the VCP, the START command must be written in VCPEXE. Writing a START stops any ongoing activity, generates a VCPXEVT, and the VCP waits for input control parameters.

12.2 Stop

To stop the VCP, the STOP command must be written in VCPEXE. The VCP stops any ongoing activity and goes into an idle state (VCPSTAT0 = 0).

12.3 Pause

To pause the VCP, the PAUSE command must be written in VCPEXE. Writing a PAUSE pauses the processing unit. Any ongoing EDMA transfer runs to completion but no subsequent event is generated. The PAUSE command is acknowledged by setting the PAUS bit in VCPSTAT0 to 1.

12.4 Unpause

To unpause the VCP, the UNPAUSE command must be written in VCPEXE. Writing an UNPAUSE unpauses the processing unit. Any event to be generated is generated. The UNPAUSE command is acknowledged by clearing the PAUS bit in VCPSTAT0 to 0.

13 Errors and Status

An error occurs if the VCP receives an invalid value in the input configuration parameters. If an error is detected, the VCPERR bit field is set accordingly, the ERR bit in VCPSTAT0 is set to 1, the VCPINT interrupt is generated, and no processing is engaged. The only way to restart the VCP is to read VCPERR and send another START command. VCPINT has an interrupt selector value of 30. See the *TMS320C6000 DSP Interrupt Selector Reference Guide* (SPRU646) for details on how to setup interrupts.

The status registers are provided for debugging purposes and are best used when either the processor is halted or the VCP is halted. If an error occurs, the VCP is halted and a VCPINT interrupt is generated that can be mapped to a CPU interrupt. There may be cases where you would want to view the status registers when the VCP is still running. One such case is when the VCP seems to have taken a long time in processing the current frame. In such cases, a watchdog timer should be used and set according to the frame length and VCP configuration in addition to some overhead to allow for EDMA usage.

14 Performance

14.1 Cycles

The state metric unit and the traceback unit can work in parallel provided the frame has been segmented in multiple sliding windows. There is also a degree of parallelism inside those units as highlighted in Table 25 and Table 26 (the VCP runs at the CPU clock divided by 4).

The processing unit performance is shown in Table 27. The overall processing delay caused by the VCP are given by the maximum between the I/O delay (EDMA transfers are at a maximum rate of 64 bits at a frequency of CPU clock divided by 4) and the actual processing unit processing delay.

Table 25. State Metric Unit Parallelism

K	Cycles/Symbols (VCP cycles)	Required Radix-2 ACS/Symbol	Cycles/Radix-2 ACS (VCP cycles)
9	32.0 (256/8)	128	0.25
8	18.9 (132/7)	64	0.29
7	12.0 (72/6)	32	0.37
6	8.40 (42/5)	16	0.52
5	4.25 (17/4)	8	0.53

Table 26. Traceback Unit Parallelism

K	VCP Cycles/Symbols
9	0.25 (2/8)
8	0.29 (2/7)
7	0.34 (2/6)
6	0.40 (2/5)
5	0.25 (1/4)

Table 27. VCP Processing Unit Performance (in VCP Cycles)

	Traceback Mode		
K	Tailed	Convergent	Mixed
9	$((256 + 2)/8) \times (F + K - 1)$	$(256/8) \times (F + C) + (2/8) \times (R + C)$	$(256/8) \times (F + C) + (2/8) \times (R + K - 1)$
8	$((132 + 2)/7) \times (F + K - 1)$	$(132/7) \times (F + C) + (2/7) \times (R + C)$	$(132/7) \times (F + C) + (2/7) \times (R + K - 1)$
7	$((72 + 2)/6) \times (F + K - 1)$	$(72/6) \times (F + C) + (2/6) \times (R + C)$	$(72/6) \times (F + C) + (2/6) \times (R + K - 1)$
6	$((42 + 2)/5) \times (F + K - 1)$	$(42/5) \times (F + C) + (2/5) \times (R + C)$	$(42/5) \times (F + C) + (2/5) \times (R + K - 1)$
5	$((17 + 1)/4) \times (F + K - 1)$	$(17/4) \times (F + C) + (1/4) \times (R + C)$	$(17/4) \times (F + C) + (1/4) \times (R + K - 1)$

Legend: F = Frame size, C = convergence distance, K = constraint length.

14.2 Bit Error Rate

An example of Bit Error Rate (BER) performance for tailed traceback mode with AMR 12.2 kbps class A bits frames is shown in Figure 24, and for mixed traceback mode with AMR 12.2 kbps class B bits frames is shown in Figure 25.

Figure 24. AMR 12.2 kbps Class A

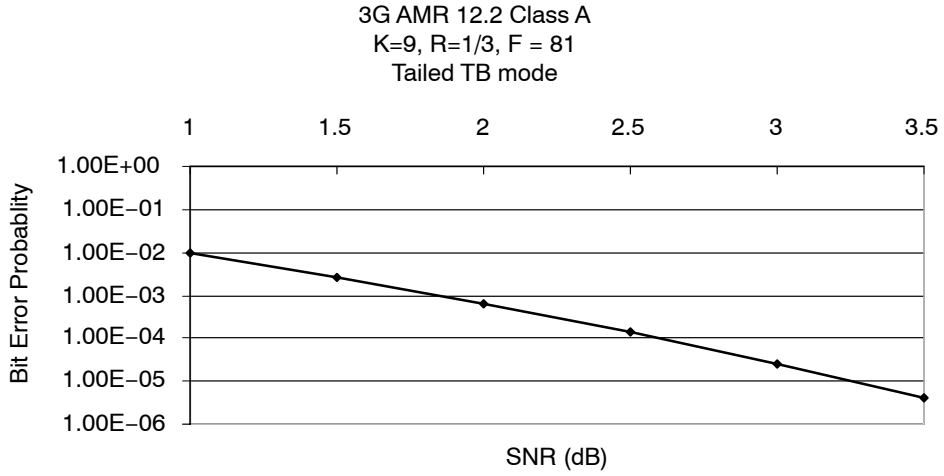
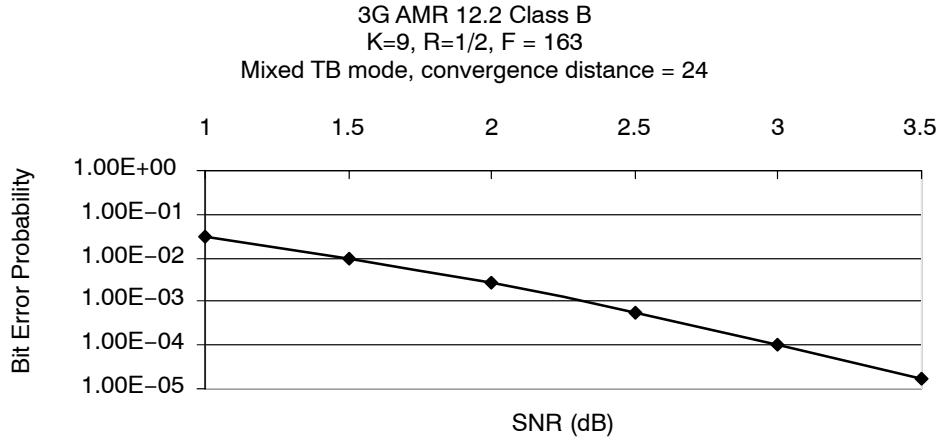


Figure 25. AMR 12.2 kbps Class B



This page is intentionally left blank.

Revision History

Table 28 lists the changes made since the previous version of this document.

Table 28. Document Revision History

Page	Additions/Modifications/Deletions
25	Changed BM bit description for value = 1 in Table 14 to: Native format (8 bits)
47	Added last two sentences in first paragraph of Section 13: VCPINT has an interrupt selector value of 30. See the *TMS320C6000 DSP Interrupt Selector Reference Guide* (SPRU646) for details on how to setup interrupts.
47	Added second paragraph of Section 13.

This page is intentionally left blank.

Index

O

P

R

S

T

V

W

Y