TMS320C645x DSP
Viterbi-Decoder Coprocessor 2 (VCP2)

User's Guide

Literature Number: SPRU972C
April 2006–Revised December 2009

List of Figures

List of Tables

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 2 (VCP2) provided in the C645x devices has been designed to perform Viterbi decoding for IS2000 and 3GPP wireless standards. The VCP2 coprocessor has been designed to perform forward-error correction for 2G and 3G wireless systems. The VCP2 coprocessor offers a very cost effective and synergistic solution when combined with Texas Instruments (TI) DSPs. The VCP2 supports 762 12.2 Kbps 3G AMR channels when running at 333 MHz. This document describes the operation and programming of the VCP2.

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.

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.

SPRU965 — *TMS320C6455 Technical Reference.* An introduction to the TMS320C6455 DSP and discusses the application areas that are enhanced.

SPRUE52 — *TMS320C645x DSP Peripherals Overview Reference Guide.* This document lists the peripherals for the TMS320C645x family of devices.

SPRC234 — *TMS320C6455 Chip Support Libraries (CSL).* A download with the latest chip support libraries.

TMS320C645x Viterbi-Decoder Coprocessor 2

Channel decoding of voice and low bit-rate data channels found in cellular standards such as 2.5G, 3G, and WiMAX requires the decoding of convolutional encoded data. The Viterbi-decoder coprocessor 2 (VCP2) provided in the C645x devices performs Viterbi decoding for IS2000 and 3GPP wireless standards. The VCP2 coprocessor also performs forward-error correction for 2G and 3G wireless systems. The VCP2 coprocessor offers a very cost effective and synergistic solution when combined with Texas Instruments (TI) DSPs. The VCP2 supports 762 12.2 Kbps 3G AMR channels when running at 333 MHz.

1 Features

The VCP2 provides:

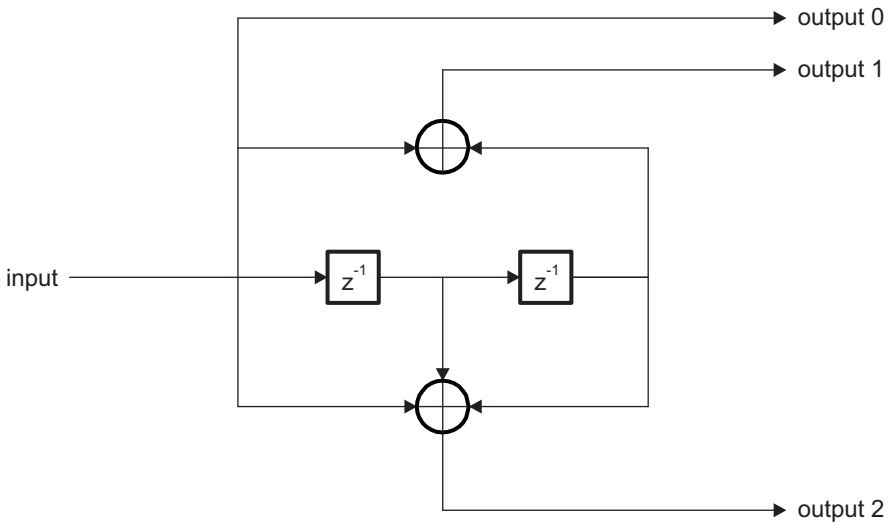
- High flexibility:
  - Variable constraint length, K = 5, 6, 7, 8, or 9
  - User-supplied code coefficients
  - Code rates (1/2, 1/3, or 1/4)
  - Configurable trace back settings (convergence distance, frame structure)
  - Branch metrics calculation and depuncturing done in software by the DSP
- System and development cost optimization:
  - The VCP2 releases DSP resources for other processing
  - Reduces board space and power consumption by performing on-chip decoding
  - Communication between the DSP and the VCP2 is performed through the high-performance EDMA3 engine
  - Uses its own optimized working memories
  - Provides debug capabilities during frame processing
  - Libraries are provided for reduced development time

2 Introduction

A convolutional code is generated by passing the information sequence to be transmitted through a linear finite-state shift register. The VCP2 is able to decode only a subset of those codes known as a 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 VCP2 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 VCP2 supports 1/2, 1/3, and 1/4 codes with rates).
- The generator polynomials Gn describe how the outputs are generated from the inputs.

Figure 1. Convolutional Encoder Example Block Diagram



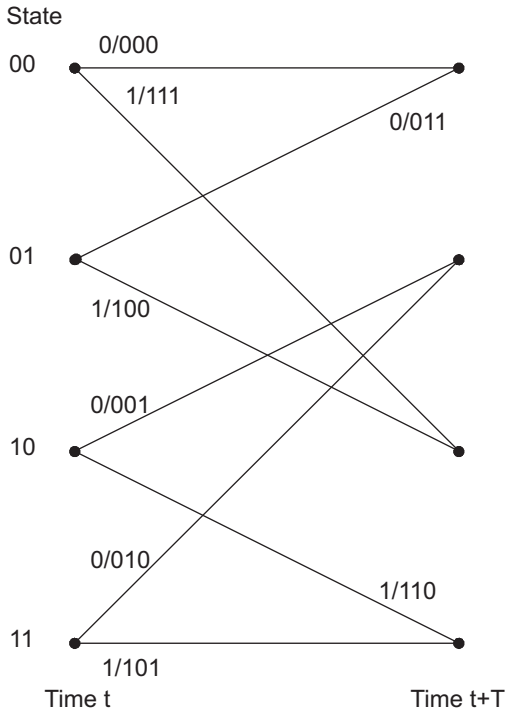
NOTE: K = 3, R = k/n = 1/3, $G_0 = (100)_8$, $G_1 = (101)_8$, $G_2 = (111)_8$ 0/000 means input is 0, output0 is 0, output1 is 0, output2 is 0.There are $2^{(K-1)}$ states and 2^k incoming branches per state.

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 VCP2 accumulates on 13 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

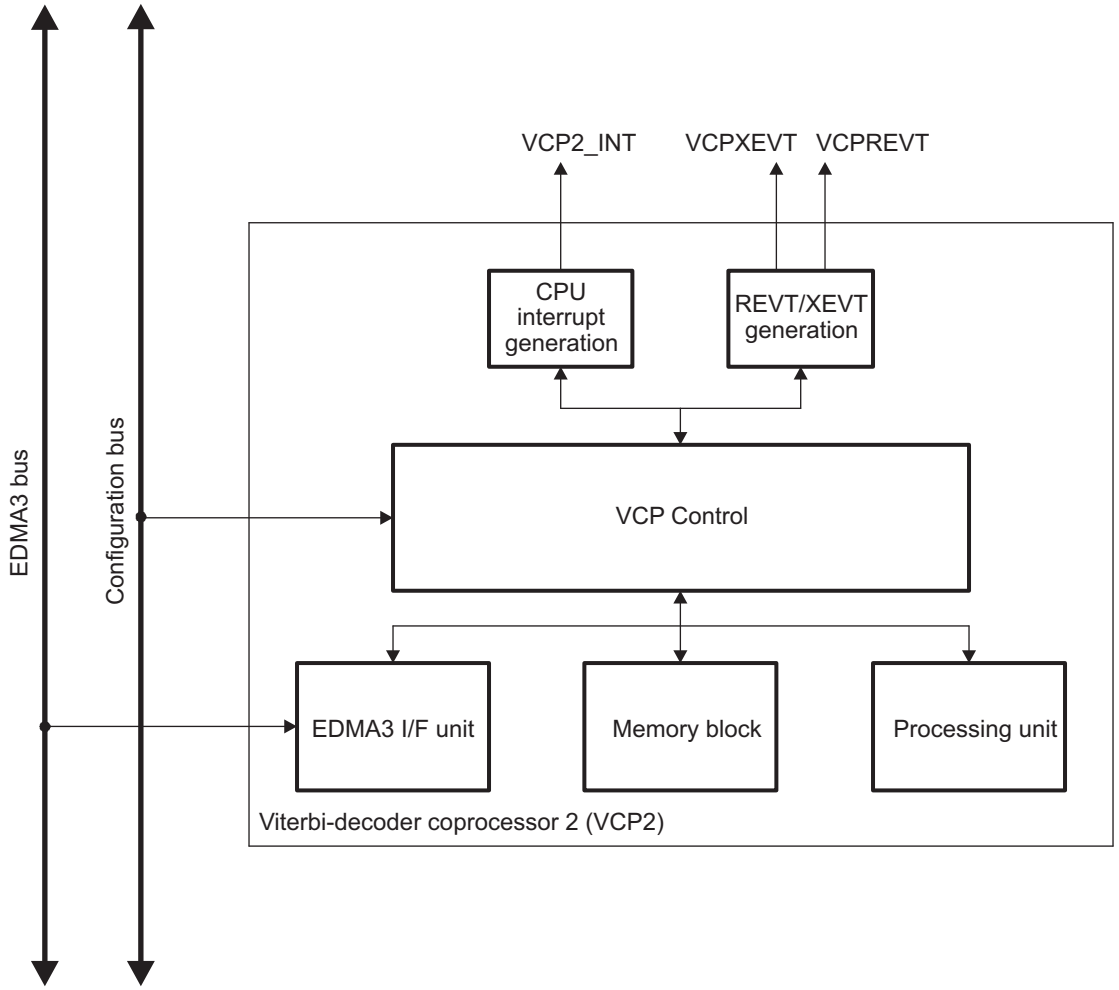


NOTE: K = 3, R = k/n = 1/3, $G_0 = (100)_8$, $G_1 = (101)_8$, $G_2 = (111)_8$ 0/000 means input is 0, output1 is 0, output2 is 0, output3 is 0. There are $2^{(K-1)}$ states and 2^k incoming branches per state.

3 Overview

The DSP controls the operation of the VCP2 (Figure 3) using memory-mapped registers. The DSP typically sends and receives data using synchronized EDMA3 transfers through the EDMA3 bus. The VCP2 sends two synchronization events to the EDMA3: a receive event (VCPREVT) and a transmit event (VCPXEVT). The VCP2 input data corresponds to the branch metrics and the output data to the hard decisions or soft decisions.

Figure 3. VCP2 Block Diagram



4 Input Data

4.1 Branch Metrics Calculations

The branch metrics (BM) are calculated by the DSP and stored in the DSP memory subsystem as 8-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 VCP2. Moreover, n soft inputs are required to calculate 1 branch metric.

Assuming BSPK modulated bits (0 → 1, 1 → -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 VCP2 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 VCP2 endian mode register (VCPEND) indicates if the data is 32-bit word packed or native 8-bit format and the VCP2 will handle the endianness byte swapping accordingly (see Section 8).

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	...			

The state metric accumulation resolution is 13 bits on the VCP2. Consequently, full 8-bit dynamic range is available for branch metrics on the C645x VCP2, for all constraint lengths and all code rates.

4.2 Soft Input Dynamic Ranges

The VCP2 implementation implies that the soft inputs need to be quantized so that the branch metrics satisfy the following bound B1 (branch metrics upper bound - absolute value):

$2^{(C - 1)} - 1 \geq (2 \times (K - 1) + 2) \times B_1$

K is the constraint length and *C* determines the truncation of state metrics that can be performed without loss of decoding performance.

The VCP2 is designed with C = 13. The branch metrics can have a maximum dynamic range of 7 + 1 sign bits [-128; +127]. This gives another branch metrics upper bound $B_2 \leq 128$.

So for a given constraint length, min (B_1, B_2) gives the final branch metrics maximum bound B.

To satisfy B in the branch metrics calculation, the soft input values, delivered as 8-bit-signed equalized values, are linearly scaled with the following formula where 1/n is the rate.

Scaled = min (B_1, B_2)/n × *SoftValue*/128

Example

K = 9, then $B_1 \leq 227.5$ and the branch metrics range B_2 is [-128; +127]. So the branch metrics need to be in [-128;+127] range.

If rate 1/3, 128/3 42, so the soft inputs need to be scaled by a factor of 0.333333 and saturated within the range [-42; +42].

Table 4 summarizes the calculations for the different constraint length and rate:

Table 4. VCP2 Soft Inputs Quantization

1/Rate	K	Scaling Factor	Range
2	5, 6, 7, 8, 9	0.5	[-64; +63]
3	5, 6, 7, 8, 9	0.333333	[-42; +42]
4	5, 6, 7, 8, 9	0.25	[-31; +31]

5 VCP2 Memory Sleep Mode

All VCP2 memories can be put into a low leakage (sleep) state. Putting the banks into a sleep mode disables the memories. The VCP2 can put the banks into dynamic sleep mode if the input control bits slpzvdd_en and slpzvss_en in the Endianness Register are set. Once the bits are set, VCP2 memories will be put in the sleep mode based on Table 5:

Table 5. VCP2 Memory Sleep Mode Truth Table

RAM	Conditions
sd	If (sdhd = 0), bank is in sleep mode. Else, bank wake up.
sm	If (exc_cmd = 0), bank is in sleep mode. Else, bank wake up.

6 Decision Data

The VCP2 can be configured to generate either hard decisions (one bit per decision), or soft decisions (8-bit value per decision). Ordering of the VCP2 decisions depends on the OUT_ORDER field of VCPIC3 and the SD field of VCPEND. If the DSP is set to work in big-endian mode and the results are soft decisions (see the VCP2 endian mode register, Section 7.3). The decisions buffer start address must be double-word aligned and the buffer size must be a multiple of 8 bytes.

The soft decisions in the VCP2 are initially computed with the path metrics at 13-bit values. The results are then clipped to 8-bit signed integer values before being stored in the traceback soft decision memory.

7 Registers

The VCP2 contains several memory-mapped registers accessible by the CPU, the IDMA, the QDMA, and the EDMA3. A configuration-bus access is faster than an EDMA3-bus access for isolated accesses (typically when accessing control registers). EDMA3-bus accesses are used for EDMA3 transfers and provide maximum throughput to/from the VCP2. The registers are listed in Table 6. For the memory map and full register addresses, see the device-specific data manual.

The branch metric and traceback 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 6. VCP2 Registers

EDMA3 Bus Offsets	Configuration Bus Offsets	Acronym	Register Name	See
	0000h	VCPPID	VCP peripheral ID register	Section 7.1
0000h		VCPIC0	VCP input configuration register 0	Section 7.2
0004h		VCPIC1	VCP input configuration register 1	Section 7.3
0008h		VCPIC2	VCP input configuration register 2	Section 7.4
000Ch		VCPIC3	VCP input configuration register 3	Section 7.5
0010h		VCPIC4	VCP input configuration register 4	Section 7.6
0014h		VCPIC5	VCP input configuration register 5	Section 7.7
0048h		VCPOUT0	VCP output register 0	Section 7.8
004Ch		VCPOUT1	VCP output register 1	Section 7.9
0080h		VCPWBM	VCP branch metrics write FIFO register	
00C0h		VCPRDECS	VCP decisions read FIFO register	
	0018h	VCPEXE	VCP execution register	Section 7.10
	0020h	VCPEND	VCP endian mode register	Section 7.11
	0040h	VCPSTAT0	VCP status register 0	Section 7.12
	0044h	VCPSTAT1	VCP status register 1	Section 7.13
	0050h	VCPERR	VCP error register	Section 7.14
	0060h	VCPEMU	VCP emulation control register	Section 7.15

Table 7. VCP2 Memories

EDMA3 Bus Offsets	Acronym	Name	Size
1000h	BM	Branch Metrics (BM)	256 Bytes
2000h	SM	State Metric (SM)	448 Bytes
3000h	TBHD	Traceback Hard Decision	4K Bytes
6000h	TBSD	Traceback Soft Decision	16K Bytes
F000h	IO	Decoded Bits (IO)	512 Bytes

NOTE: Register and Memory Access

- Data Transfer Alignment: Normal (non-emulation) mode data transfers to/from the VCP2 must be aligned on a double-word (64-bit) boundary. Alignment can be forced in C using the 'DATA_ALIGN' pragma. Non-alignment results in data transfer failure.

  Example:

```
#pragma DATA_ALIGN(configIc, 8)  // Should be double-word aligned
VCP_ConfigIc configIc;           // VCP Input Configuration Reg
```

- Data Transfer Size: Normal (non-emulation) mode data transfers to/from the VCP2 must be of a length that is an 8-byte (double-word) multiple.
- Emulation mode transfers are performed on 32-bit boundaries and are 4 bytes in length.

7.1 VCP2 Peripheral Identification Register (VCPPID)

The VCP2 peripheral identification register (VCPPID) is a constant register that contains the ID and ID revision number for the 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 VCPPID register is shown in Figure 4 and described in Table 8.

Figure 4. VCP2 Peripheral ID Register (VCPPID)

31 24	23 16	15 8	7 0
Reserved	TYPE	CLASS	REV
R-0	R-0x01	R-0x11	R-rev

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

Table 8. VCP2 Peripheral ID Register (VCPPID) Field Descriptions

Bit	Field	Value	Description
31-24	Reserved	0	Reserved
23-16	TYPE	01h	Peripheral Type. Identifies the type of the peripheral.
15-8	CLASS	11h	Peripheral Class. Identifies the class.
7-0	REV	<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.

7.2 VCP2 Input Configuration Register 0 (VCPIC0)

The VCP2 input configuration register 0 (VCPIC0) is shown in Figure 5 and described in Table 9.

Figure 5. VCP2 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; R = Read only; -*n* = value after reset

Table 9. VCP2 Input Configuration Register 0 (VCPIC0) Field Descriptions

Bit	Field	Value	Description[1]
31-24	POLY3	0-FFh	Polynomial generator G_3 (see Section 10.2).
23-16	POLY2	0-FFh	Polynomial generator G_2 (see Section 10.2).
15-8	POLY1	0-FFh	Polynomial generator G_1 (see Section 10.2).
7-0	POLY0	0-FFh	Polynomial generator G_0 (see Section 10.2).

(1) 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 is the least significant bit (b_0 is not passed, but set by the internal VCP hardware). The VCP2 uses the number of poly fields set to zero starting at POLY3 to determine the code rate. Therefore, POLY*n* fields not used by the current code rate must be set to zero. The VCP2 uses the number of least-significant bits that are zero in POLY0 to determine the constraint length.

7.3 VCP2 Input Configuration Register 1 (VCPIC1)

The VCP2 input configuration register 1 (VCPIC1) is shown in Figure 6 and described in Table 10.

Figure 6. VCP2 Input Configuration Register 1 (VCPIC1)

31	29	28	27	16
Reserved		YAMEN	YAMT	
R/W-0		R/W-0	R/W-0	

15	0
Reserved	
R/W-0	

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

Table 10. VCP2 Input Configuration Register 1 (VCPIC1) 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	YAMEN		Yamamoto algorithm enable bit. See Section 9.2.
		0	Yamamoto algorithm is disabled.
		1	Yamamoto algorithm is enabled.
27-16	YAMT	0-FFFh	Yamamoto threshold value bits. See Section 9.2.
15-0	Reserved	0	Reserved. These reserved bit locations must be 0. A value written to this field has no effect.

7.4 VCP2 Input Configuration Register 2 (VCPIC2)

The VCP2 input configuration register 2 (VCPIC2) is shown in Figure 7 and described in Table 11.

Figure 7. VCP2 Input Configuration Register 2 (VCPIC2)

31	16	15	0
R		FL	
R/W-0		R/W-0	

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

Table 11. VCP2 Input Configuration Register 2 (VCPIC2) Field Descriptions

Bit	Field	Value	Description
31-16	R	0-FFFFh	Reliability length bits (see Section 9.1).
15-0	FL	0-FFFFh	Frame length bits (see Section 9.1). The total number of non-encoded bits (i.e., the number of information bits that are input to the encoder at the transmitter or output from the decoder at the receiver).

7.5 VCP2 Input Configuration Register 3 (VCPIC3)

The VCP2 input configuration register 3 (VCPIC3) is shown in Figure 8 and described in Table 12.

Figure 8. VCP2 Input Configuration Register 3 (VCPIC3)

31 29	28	27 25	24	23 16
Reserved	OUT_ ORDER	Reserved	ITBEN	ITBI
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

15 0
C
R/W-0

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

Table 12. VCP2 Input Configuration Register 3 (VCPIC3) 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	OUT_ORDER	0 and 1	Defines the order of VCP output for decoded data.
		0	0 to 31 (LSB is oldest)
		1	31 to 0 (MSB is oldest)
27-25	Reserved	0	Reserved. The reserved bit location is always read as 0. A value written to this field has no effect.
24	ITBEN	0 and 1	Traceback state index enable/disable.
		0	Disabled
		1	Initialization of traceback starting state is enabled
23-16	ITBI	0-FFh	Traceback state index. The index of the starting state for the traceback unit.
15-0	C	0-FFFFh	Convergence distance bits. The length of the convergent section of the siding window. This is only used if f > F + (K-1) in mixed mode, or if f > F + C in convergence mode (see Section 9.1.4).

7.6 VCP2 Input Configuration Register 4 (VCPIC4)

The VCP2 input configuration register 4 (VCPIC4) is shown in Figure 9 and described in Table 13.

Figure 9. VCP2 Input Configuration Register 4 (VCPIC4)

31	29	28	16
Reserved		IMINS	
R/W-0		R/W-0	

15	13	12	0
Reserved		IMAXS	
R/W-0		R/W-0	

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

Table 13. VCP2 Input Configuration Register 4 (VCPIC4) 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.
28-16	IMINS	0-1FFFh	Minimum initial state metric (13 bits).
15-13	Reserved	0	Reserved. The reserved bit location is always read as 0. A value written to this field has no effect.
12-0	IMAXS	0-1FFFh	Maximum initial state metric (13 bits).

7.7 VCP2 Input Configuration Register 5 (VCPIC5)

The VCP2 input configuration register 5 (VCPIC5) is shown in Figure 10 and described in Table 14.

Figure 10. VCP2 Input Configuration Register 5 (VCPIC5)

31	30	29 28	27 25	24 20	19 16
SDHD	OUTF	TB	Reserved	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; R = Read only; *-n* = value after reset

Table 14. VCP2 Input Configuration Register 5 (VCPIC5) Field Descriptions

Bit	Field	Value	Description
31	SDHD		Output decision type select bit.
		0	Hard decisions
		1	Soft decisions
30	OUTF		Output parameters read flag bit.
		0	VCPREVT is not generated by VCP for output parameters read
		1	VCPREVT generated by VCP for output parameters read
29-28	TB		Traceback mode select bits.
		0	Not allowed
		1h	Tailed, $F \leq F_{max}$ (1)
		2h	Convergent, (no tail bits)
		3h	Mixed, $F \geq F_{max}$ and tail bits are used
27-25	Reserved	0	Reserved. The reserved bit location is always read as 0. A value written to this field has no effect.
24-20	SYMR	0-1Fh	Determines decision buffer length in output FIFO. For information on selecting the appropriate SYMR value, see Section 9.4.
19-16	SYMX	0-Fh	Determines branch metrics buffer length in input FIFO. For information on selecting the appropriate SYMX value, see Section 9.3.
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	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 are initialized with the value in the IMINS bits.

(1) For more details on F_{max}, see Section 9.1.4.

7.8 VCP2 Output Register 0 (VCPOUT0)

The VCP2 output register 0 (VCPOUT0) is shown in Figure 11 and described in Table 15.

Figure 11. VCP2 Output Register 0 (VCPOUT0)

31	29	28	16
Reserved		FMINS	
R/W-0		R/W-0	

15	13	12	0
Reserved		FMAXS	
R/W-0		R/W-0	

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

Table 15. VCP2 Output Register 0 (VCPOUT0) 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.
28-16	FMINS	0-FFFh	Minimum final state metric value (13 bits).
15-13	Reserved	0	Reserved. The reserved bit location is always read as 0. A value written to this field has no effect.
12-0	FMAXS	0-FFFh	Maximum state metric value for the final trellis stage (at trellis stage R+C). 13 bits

7.9 VCP2 Output Register 1 (VCPOUT1)

The VCP2 output register 1 (VCPOUT1) is shown in Figure 12 and described in Table 16.

Figure 12. VCP2 Output Register 1 (VCPOUT1)

31	17	16
Reserved		YAM
R/W-0		R-0

15	8	7	0
Reserved		FMAXI	
R/W-0		R-0	

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

Table 16. VCP2 Output Register 1 (VCPOUT1) Field Descriptions

Bit	Field	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. This bit is a quality indicator bit and is only used if the Yamamoto logic is enabled. See Section 9.2.
		0	At least one trellis stage had an absolute difference less than the Yamamoto threshold and the decided frame has poor quality
		1	No trellis stage had an absolute difference less than the Yamamoto threshold and the frame has good quality
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	FMAXI	0-FFFh	State index for the state with the final maximum state metric. There are $2^{(k-1)}$ state metrics for each trellis stage. Valid range for FMAXI is 0 to $2^{(k-1)}$ -1.

7.10 VCP2 Execution Register (VCPEXE)

The VCP2 execution register (VCPEXE) is shown in Figure 13 and described in Table 17.

Figure 13. VCP2 Execution Register (VCPEXE)

31	4	3	0
Reserved		COMMAND	
R/W-0		W-0	

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

Table 17. VCP2 Execution Register (VCPEXE) Field Descriptions

Bit	Field	Value	Description
31-4	Reserved	0	Reserved. The reserved bit location is always read as 0. A value written to this field has no effect.
3-0	COMMAND		VCP command select bits; see Section 11.
		0	Reserved (no instruction)
		1h	Start VCP (normal mode)
		2h	Halt or Pause VCP (debug mode). The VCP is halted (or paused) after processing the state metric for the current sliding window and before the start of the traceback.
		3h	Restart VCP and process one sliding window (debug mode). The VCP is restarted from the pause state and begins the traceback operation. The VCP is again paused after processing the state metrics for next sliding window.
		4h	Restart VCP (debug mode). The VCP is restarted from the paused state and begins the traceback operation. The VCP will run to normal completion.
		5h	Stop. Soft reset all VCP registers to their initial condition. All registers in the VCP are reset in this mode except for the execution register, endian register, emulation register, and other internal registers.
		6h-FFh	Reserved

7.11 VCP2 Endian Mode Register (VCPEND)

The VCP2 endian mode register (VCPEND) is shown in Figure 14 and described in Table 18. VCPEND has an effect only in big-endian mode.

Figure 14. VCP2 Endian Mode Register (VCPEND)

31	16
Reserved	
R/W-0	

15	10	9	8
Reserved		SLPZVSS_EN	SLPZVDD_EN
R/W-0		R/W-1	R/W-1

7	2	1	0
Reserved		SD	BM
R/W-0		R/W-0	R/W-0

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

Table 18. VCP2 Endian Mode Register (VCPEND) Field Descriptions

Bit	Field	Value	Description
31- 10	Reserved	0	Reserved. The reserved bit location is always read as 0. A value written to this field has no effect.
9	SLPZVSS_EN		Sleep mode for SLPZVSS_EN.
		0	Disable sleep mode
		1	Enable internal control of SLPZVSS
8	SLPZVDD_EN		Sleep mode for SLPZVDD_EN.
		0	Disable sleep mode
		1	Enable internal control of SLPZVDD
7-2	Reserved	0	Reserved. The reserved bit location is always read as 0. A value written to this field has no effect.
1	SD		Traceback soft-decision memory format select bit.
		0	32-bit-word packed.
		1	Native format (8 bits).
0	BM		Branch metrics memory format select bit.
		0	32-bit-word packed.
		1	Native format (8 bits).

7.12 VCP2 Status Register 0 (VCPSTAT0)

The VCP2 status register 0 (VCPSTAT0) is shown in Figure 15 and described in Table 19.

Figure 15. VCP2 Status Register 0 (VCPSTAT0)

31	29	28	16
Reserved		NSYMPROC	
R/W-0		R-0	

15	12	11	8
NSYMPROC		Reserved	
R-0		R/W-0	

7	6	5	4	3	2	1	0
Reserved	EMU HALT	OFFUL	IFEMP	WIC	ERR	RUN	PAUSE
R/W-0	R-0	R-0	R-1	R-0	R-0	R-0	R-0

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

Table 19. VCP2 Status Register 0 (VCPSTAT0) 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.
28-12	NSYMPROC		Number of symbols processed bits. The NSYMPROC bits indicate how many symbols have been processed in the state metric unit with respect to time. The maximum number of processed stages is equal to *f* + (*k*-1) in tailed or mixed mode. The maximum number of processed stages is equal to *f* + c in convergent mode.
11-7	Reserved	0	Reserved. The reserved bit location is always read as 0. A value written to this field has no effect.
6	EMUHALT		Emulation halt status bit.
		0	No halt due to emulation.
		1	Halt due to emulation.
5	OFFUL		Output FIFO buffer full status bit.
		0	Output FIFO buffer is not full.
		1	Output FIFO buffer is full.
4	IFEMP		Input FIFO buffer empty status bit.
		0	Input FIFO buffer is not empty.
		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.
		0	Not waiting for input configuration words.
		1	Waiting for input configuration words.
2	ERR		VCP error status bit. The ERR bit is cleared as soon as the DSP reads the VCP error register (VCPERR).
		0	No error.
		1	VCP paused due to error.
1	RUN		VCP running status bit.
		0	VCP is not running.
		1	VCP is running.
0	PAUSE		VCP pause status bit.
		0	VCP is not paused. The UNPAUSE command is acknowledged by clearing the PAUSE bit.
		1	VCP is paused. The PAUSE command is acknowledged by setting the PAUSE bit. The PAUSE bit can also be set, if the input FIFO buffer is becoming empty or if the output FIFO buffer is full.

7.13 VCP2 Status Register 1 (VCPSTAT1)

The VCP2 status register 1 (VCPSTAT1) is shown in Figure 16 and described in Table 20.

Figure 16. VCP2 Status Register 1 (VCPSTAT1)

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

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

Table 20. VCP2 Status Register 1 (VCPSTAT1) Field Descriptions

Bit	Field	Value	Description
31-16	NSYMOF	0-FFFFh	Number of symbols in the output FIFO buffer.
15-0	NSYMIF	0-FFFFh	Number of symbols in the input FIFO buffer.

7.14 VCP2 Error Register (VCPERR)

The VCP2 error register (VCPERR) is shown in Figure 17 and described in Table 21.

Figure 17. VCP2 Error Register (VCPERR)

31							8
Reserved							
R-0							
7	6	5	4	3	2	1	0
Reserved	E_SYMR	E_SYMX	MAXMIN ERR	FCTLERR	FTLERR	TBNAERR	ERROR
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 21. VCP2 Error Register (VCPERR) Field Descriptions

Bit	Field	Value	Description
31-7	Reserved	0	Reserved. The reserved bit location is always read as 0. A value written to this field has no effect.
6	E_SYMR	0	No error for SMAR, as shown in the following relationships:
			TB Mode / **SYMR** / **# of 64 Bit Transfers** / **Condition Transfer** Hard / 0 to 31 / 1 to 32 / f ≤ 2048 Hard / 31 / 32 / f > 2048 Hard / 15 / 16 / f ≥ 2048 Soft / 0 to 31 / 1 to 32 / f ≤ 2048 Soft / 31 / 32 / f ≥ 2048 Soft / 15 / 16 / f > 2048
		1	Error occurred
5	E_SYMX	0	No error for SMAX, as shown in the following relationships:
			Code Rate / **SYMX** / **# of 64 Bit Transfers** / **# of BM per** / **# trellis per** 1/4 / 3 / 16 / 128 / 16 1/4 / 1 / 8 / 64 / 8 1/3 / 7 / 16 / 128 / 32 1/3 / 3 / 8 / 64 / 16 1/2 / 15 / 16 / 128 / 64 1/2 / 7 / 8 / 64 / 32
		1	Error occurred
4	MAXMINERR	0	No error. Error check for the following relationships:
			Sign IMAXS / **Sign IMINS** / **Check Equation** 0 / 0 / 0 ≤ IMAXS - IMINS < 2048 1 / 1 / 0 ≤ IMAXS - IMINS < 2048 0 / 1 / 0 ≤ IMAXS - IMINS < 2048 1 / 0 / 0 ≤ (2 × 4096 + IMAXS) - IMINS < 2048
		1	Error occurred
3	FCTLERR	0	No error
		1	1 = *r* + *c* too large [(*r* + c) > (*r* + c)max] for mixed or convergent traceback modes
2	FTLERR	0	No error
		1	F too large (*f* > F_{max}) for tailed traceback mode(1).
1	TBNAERR	0	No error
		1	Traceback mode is not allowed. (1 to 3 are valid modes)

(1) For more details on F_{max}, see Section 9.1.4.

Table 21. VCP2 Error Register (VCPERR) Field Descriptions (continued)

Bit	Field	Value	Description
0	ERROR	0	No error is detected.
		1	An error has occurred

7.15 VCP2 Emulation Control Register (VCPEMU)

The VCP2 emulation control register (VCPEMU) is shown in Figure 18 and described in Table 22.

Figure 18. VCP2 Emulation Control Register (VCPEMU)

31	2	1	0
Reserved		SOFT	FREE
R/W-0		R/W-0	R/W-0

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

Table 22. VCP2 Emulation Control Register (VCPEMU) 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	(Default mode) VCP completes the current window of state metric processing and halts before starting traceback or at the end of a frame
		1	VCP completes a frame of data before halting
0	FREE		Free bit
		0	Reserved
		1	Free run mode - vcp_emususp signal and functions normally.

8 Endianness

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, data interpretation will be different.

8.1 Branch Metrics

When the data are saved in their 8-bit native format (BM = 1), they must be organized in the DSP memory as described in Table 24. When the data are packed on 32-bit words (BM = 0), they must be organized in the DSP memory as described in Table 25.

Table 23. Branch Metrics

Little_big_endian	BM	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)

Table 24. 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

Data are presented to the EDMA3 as shown in Figure 19. The endianness manager reorders the BMs, as shown in Figure 20, for processing.

Figure 19. Data Source - VBUSP/DMA (BM = 1)

63:56	55:48	47:40	39:32	31:24	23:16	15:8	7:0
BM0	BM1	BM2	BM3	BM4	BM5	BM6	BM7

Figure 20. Data Destination - Kernel for Processing Unit (BM = 1)

63:56	55:48	47:40	39:32	31:24	23:16	15:8	7:0
BM7	BM6	BM5	BM4	BM3	BM2	BM1	BM0

Table 25. 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

Data are presented to the EDMA3 as shown in Figure 21. The endianness manager reorders the BMs, as shown in Figure 22, for processing.

Figure 21. Data Source - VBUSP/DMA (BM = 0)

63:56	55:48	47:40	39:32	31:24	23:16	15:8	7:0
BM3	BM2	BM1	BM0	BM7	BM7	BM5	BM4

Figure 22. Data Destination - Kernel for Processing Unit (BM = 0)

63:56	55:48	47:40	39:32	31:24	23:16	15:8	7:0
BM7	BM6	BM5	BM4	BM3	BM2	BM1	BM0

8.1.1 Hard Decisions

The VCP2 hard-decisions bit ordering within the 32-bit hard-decision word is programmable via the OUT_ORDER = 0 register, such that the oldest bit can be either in the MSB or the LSB position.

Figure 23. 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

OUT_ORDER = 1 orders the hard-decision data in the order it is calculated in the state metric computation, from 31 to 0 in the 32-bit word output.

Figure 24. Trellis Stage 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

8.1.2 Soft Decisions

The VCP2 soft decisions are 8-bit results and output 64 bits at a time. The soft decisions are organized as shown in Table 26, based on the CPU's endianness and whether SD is set for native or 32-bit packed results.

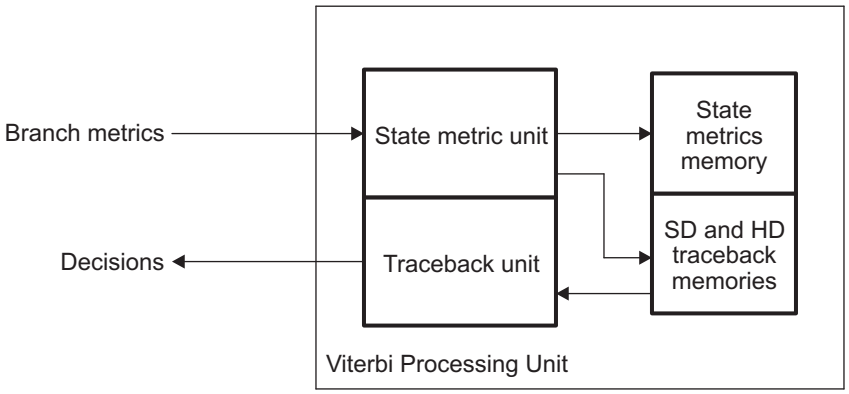
Table 26. Soft Decision Organization

Little_big_endian	BM	Description (MSB to LSB)
0	0	7,6,5,4,3,2,1,0 ⇒ 3, 2, 1, 0, 7, 6, 5, 4 (bytes)
0	1	7,6,5,4,3,2,1,0 ⇒ 0, 1, 2, 3, 4, 5, 6, 7 (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)

9 Architecture

The VCP2 processing unit is shown in Figure 25. 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 25. Processing Unit



9.1 Sliding-Windows Processing

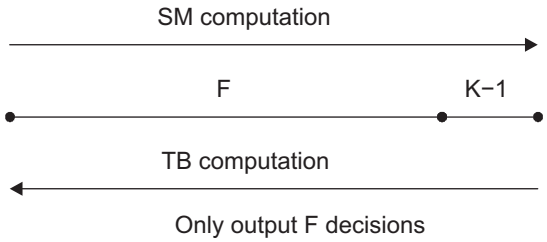
The traceback hard-decision memory can store up to 32768 traceback bits and there are $2^{(K-1)}$ bits stored at each trellis stage. Therefore, the traceback hard-decision memory can store decisions of $32768/2^{(K-1)}$ symbols. The traceback soft-decision memory can store up to 8192 traceback soft values 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 the decisions do not fit in the traceback HD/SD memory, then the convergent or mixed mode is used and the original frame is segmented into sliding windows (SW); otherwise, the traceback mode is set to tailed and no segmentation is required.

9.1.1 Tailed Traceback Mode

This mode is used when the frame is terminated and fits within the coprocessor traceback memory (see Figure 26). The state metrics are computed over $F + K - 1$ symbols, the traceback is initialized to the proper 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.

Figure 26. Tailed Traceback Mode



9.1.2 Mixed Traceback Mode

This mode is used when the frame is terminated and does not fit within the coprocessor traceback memory. The frame is split into sliding windows (see Figure 27). The state metrics are computed over *F* + *K* - 1 symbols, the traceback is initialized to the proper 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 9.3). 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.

9.1.3 Convergent Traceback Mode

This mode is used with non-terminated 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 28). The state metrics are computed over *F* + *C* symbols, the traceback is initialized to the proper 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 9.4). The state metrics computation of sliding window *I* + 1 is done in parallel with the traceback computation of sliding window *I*.

Figure 27. Mixed Traceback Mode-Example With Five Sliding Windows

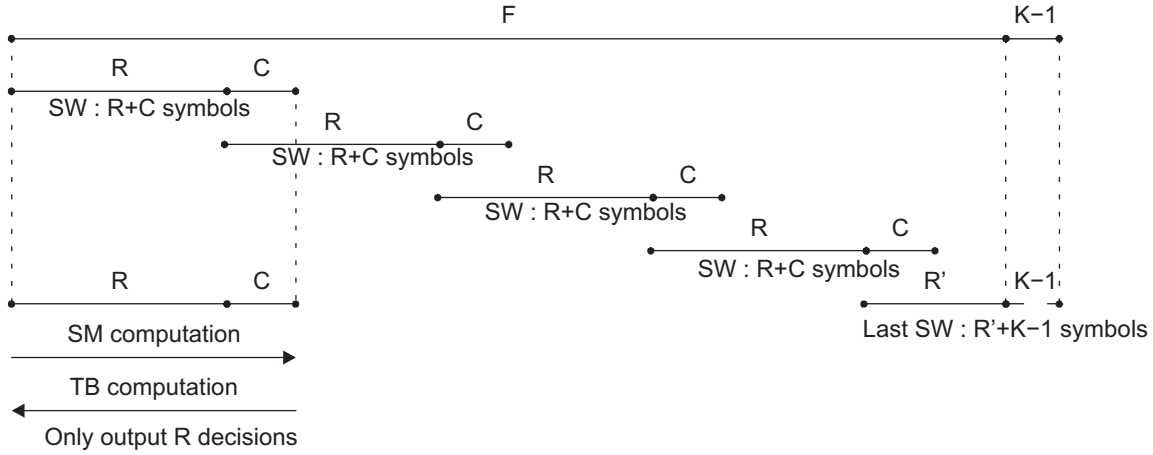
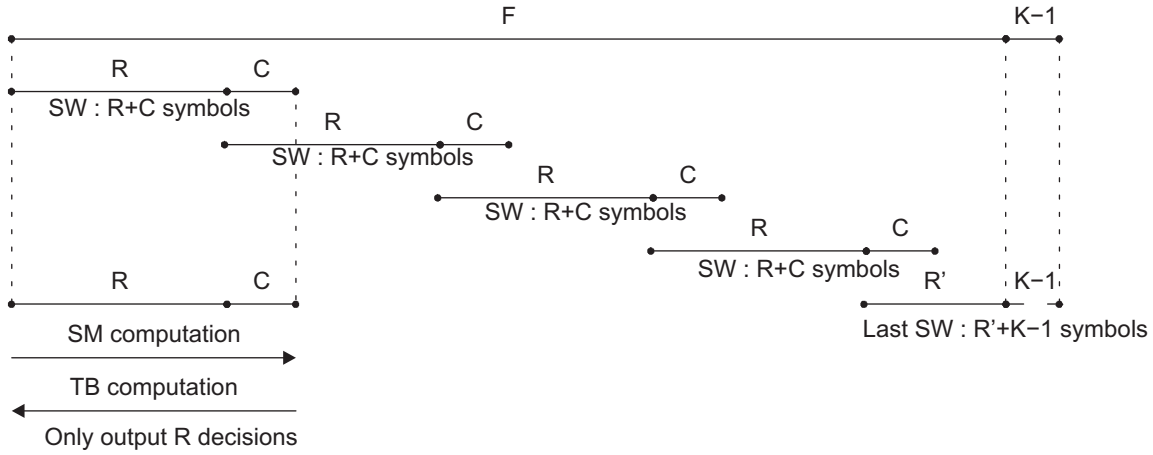


Figure 28. Convergent Traceback Mode-Example With Five Sliding Windows



9.1.4 F, R, and C Limitations

VCP2 has increased the traceback soft-decision memory and output FIFO compared to the memory in VCP. The traceback soft-decision memory size has been increased from 1024 × 96 bits to 2048 × 64 bits. The output FIFO memory has been increased from 32 × 64 bits to 64 × 64 bits. In addition, the soft-decision resolution is increased internally to 13 bits and is clipped to 8 bits when output from VCP2. These memory sizes have an effect on the F_{max} parameter (i.e., the max. frame size for tailed processing), and the $(R+C)_{max}$ parameter (i.e., the max. sliding window length for mixed/convergent processing).

The differences between VCP and VCP2 are related to F_{max}, $(R+C)_{max}$, and C. The impact on VCP2 performance is as follows:

- Allowed values for C are C = N*(K - 1), where N is a convergence multiplier. Table 27 and Table 28 show the possible values of N for each value of K for hard and soft decisions. The larger the convergence length is, the better the BER. However, larger convergence lengths require more VCP clock cycles.
- Soft decisions:
  - R is constrained to be less than or equal to 248.
  - F_{max} and $(R+C)_{max}$ for VCP2 have been increased over VCP. This change reduces the number of sliding windows and, therefore, decreases the number of VCP2 cycles needed for traceback. Overall, there is no major improvement in the processing delay because a larger portion of cycles is spent in state metric accumulation, which is largely unaffected by the choice of R and C.
- Hard decisions:
  - $(R+C)_{max}$ for mixed/convergent processing for hard decisions has been increased from 605 to 635 for K = 6, and from 1020 to 2044 for K = 5. As with soft decisions, this change results in a small decrease in VCP2 cycle counts during traceback.

The procedure to calculate the reliability length is as follows [the reliability length cannot be larger than 1920, C ≤ 1920 - (k - 1)]:

1. Determine the convergence length C = N*(K - 1).
2. Determine the number of sliding windows: N_{sw} = ceil(f/[$(r + c)_{max}$ - c]).
3. Determine the reliability length: R = m × ceil[f/(Nsw × m)].
4. If hard decisions are being used and R > 1920 or if soft decisions are being used and R > 248, then increment the number of sliding windows and go back to step 3.
5. For hard decision and soft-decision limits, see Table 27 and Table 28.

Table 27. Traceback Soft Decision Sliding Window Limits

K	F_{max}	$(r + c)_{max}$	Multiple (m)	Convergent Multiplier (N)	Special Constraint
9	56	60	4	3,6	
8	105	105	7	3,6,9,12	
7	186	186	6	3,6,9,12,15,18	
6	315	315	5	3,6,9,12,15,18	r ≤ 248
5	1020	320	4	3,6,9,12,15,18	r ≤ 248

Table 28. Traceback Hard Decision Sliding Window Limits

K	F_{max}	$(r + c)_{max}$	Multiple (m)	Convergent Multiplier (N)
9	120	124	4	3,6,9,12,15
8	217	217	7	3,6,9,12,15,18
7	378	378	6	3,6,9,12,15,18
6	635	635	5	3,6,9,12,15,18
5	2044	2044	4	3,6,9,12,15,18

The maximum value for the FL input parameter is F_{max} for tailed mode or $(r+c)_{max}$ for mixed or convergent modes.

9.2 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).

Basically, 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; 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.

9.3 Input FIFO (Branch Metrics)

The branch metric input FIFO uses a double-buffering scheme to allow the transfer of new input data while processing the current data. After the VCP2 is initiated, it generates a VCPXEVT EDMA synchronization event each time one side of the input FIFO is empty (and, thus, ready to accept new data). The value of SYMX in the VCPIC5 register determines the number of 64-bit transfers of input data expected to be written into the input FIFO by the EDMA for each VCPXEVT event.

Table 29 lists the valid values for SYMX, along with the corresponding number of expected 64-bit transfers. As shown for the supported code rates, the VCP2 can be programmed to expect either 8 or 16 64-bit transfers for each VCPXEVT event.

The VCP2 only generates as many VCPXEVT events as needed to transfer all the branch metric input data required for the current code block. In other words, no excess VCPXEVT events are generated based on the FIFO being empty at the end of processing.

Table 29. Code Rate versus SYMX

Code Rate	SYMX	Number of 64-Bit Transfers
1/4	3	16
1/4	1	8
1/3	7	16
1/3	3	8
1/2	15	16
1/2	7	8

9.4 Output FIFO (Decisions)

The decoded decision output FIFO uses a double-buffering scheme to allow the EDMA to transfer out available decoded data while the VCP2 processes and writes more decoded data. The VCP2 generates a VCPREVT EDMA synchronization event each time one side of the output FIFO is full. In the case that all the decoded data fits within one side of the output FIFO, only one VCPREVT is generated after all the data has been written to the FIFO.

The value of SYMR in the VCPIC5 register should be set to one less than the number of 64-bit transfers of output data expected to be transferred from the output FIFO by the EDMA for each VCPREVT event. The possible range for SYMR is 1 to 31. SYMR should be calculated as follows:

- For hard decisions:
  - If F ≤ 2048, then SYMR = ceil (F/64) - 1
  - If F > 2048, then SYMR = 15 or 31
- For soft decisions:

- If F ≤ 256, then SYMR = ceil (F/8) - 1
- If F > 256, then SYMR = 15 or 31

The number of 64-bit transfers per VCPREVT event is SYMR + 1. Again, when F ≤ 2048, for hard decision output, or F ≤ 256, for soft decision output, and SYMR is calculated as shown, a single VCPREVT event is generated once all the output data has been written to the output FIFO.

10 Programming

The VCP2 requires setting up the following context per user channel:

- 3 to 4 EDMA3 parameters (see Table 30)
- The input configurations parameters

Several user channels can be programmed prior to starting the VCP2. A suggested implementation is to use the EDMA3 interrupt generation capabilities [see the *TMS320C645x DSP Enhanced DMA (EDMA3) Controller User's Guide* (SPRU966)] and program the EDMA3 to generate an interrupt after the user channel's last VCPREVT synchronized EDMA3 transfer has completed.

Table 30. Required EDMA3 Links Per User Channel

Direction[1]	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)

(1) Transmit direction (DSP → VCP), receive direction (VCP → DSP)

10.1 EDMA3 Resources

10.1.1 VCP2 Dedicated EDMA3 Resources

Within the available 64 EDMA3 channel event sources, two are assigned to the VCP2: event 28 and event 29.

- Event 28 is associated to the VCP2 receive event (VCPREVT) and is used as the synchronization event for EDMA3 transfers from the VCP2 to the DSP (receive). EDMA3 channel 28 is primarily intended to serve VCP2-to-DSP transfers.
- Event 29 is associated to the VCP2 transmit event (VCPXEVT) and is used as the synchronization event for EDMA3 transfers from the DSP to the VCP2 (transmit). EDMA3 channel 29 is primarily intended to serve DSP-to-VCP2 transfers.

10.1.2 Special VCP2 EDMA3 Programming Considerations

The EDMA parameters consist of eight words as shown in Figure 29. All EDMA transfers, in the context of the VCP, 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 ACNT for the VCP EDMA transfer must be a multiple of 8. Single-word transfers that are not double-word aligned cause errors in the TCP2/VCP2 memory.

For more information, see the *TMS320C645x DSP Enhanced DMA (EDMA3) Controller User's Guide* (SPRU966).

Figure 29. 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)

10.1.2.1 Input Configuration Parameters Transfer

This EDMA3 transfer to the input configuration parameters is a 6-word VCPXEVT frame-synchronized transfer. The OPTIONS should be set as:

- ITCCEN = 0 (Intermediate transfer complete chaining is disabled)
- TCCEN = 0 (Transfer complete chaining is disabled)
- ITCINTEN = 0 (Intermediate transfer complete interrupt is disabled)
- WIMODE = 0 (Normal operation)
- 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 transfer, each event triggers the transfer of ACNT elements)
- DAM = 0 (Dst address within an array increments. Dst is not a FIFO.)
- SAM = 0 (Src Address within an array increments. Src is not a FIFO.)
- SOURCE ADDRESS: User input configuration parameters start address
- ACNT = 24 (Number of bytes in an array)
- BCNT = 1 (Number of arrays in a frame)
- Destination Address: VCPIC0
- SRCBIDX = 0
- DSTBIDX = 0
- SRCCIDX = 0
- DSTCIDX = 0
- CCNT = 1 (Number of frames in a block)

Upon completion, this EDMA3 transfer is linked to the EDMA3 for branch metrics transfer parameters.

10.1.2.2 Branch Metrics Transfer

This EDMA3 transfer to the branch metrics FIFO is a VCPXEVT frame-synchronized transfer. The OPTIONS should be set as:

- 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)
- WIMODE = 0 (Normal operation)
- 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 transfer, each event triggers the transfer of ACNT elements)
- DAM = 1 (Dst address is fixed. Dst is a FIFO.)
- SAM = 0 (Src Address within an array increments. Src is not a FIFO.)
- SOURCE ADDRESS: Branch Metrics Array start address
- ACNT = $4\times(SYMX+1)\times2^{(r-1)}$ (Number of branch metrics bytes in an array)
- BCNT = CEIL(TNBM/ACNT) (Number of arrays in a frame)
  Where TNBM = Total Number of Branch Metrics, in bytes
  To calculate total number of branch metrics data in bytes:
  For mixed and tailed traceback mode, Total number of Branch Metrics = $(F + K - 1) \times(2^{(r - 1)})$

For convergent mode, Total number of Branch Metrics = (F + C) × ($2^{(r - 1)}$)

- DESTINATION ADDRESS: VCPWBM branch metrics FIFO address
- SRCBIDX = ACNT
- DSTBIDX = 0
- SRCCIDX = 0
- DSTCIDX = 0
- CCNT = 1 (Number of frames in a block)

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

- The DMA input configuration parameters transfer parameters of the next user channel, if there is one ready to be decoded.
- Dummy DMA transfer parameters, if there are no more user channels ready to be decoded [for information on how to set up a dummy Xfer, see the *TMS320C645x DSP Enhanced DMA (EDMA3) Controller User's Guide* (SPRU966)]. Do not link to a NULL transfer, as the secondary event register will set the event flag for Event 29. The final VCPXEVT 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 VCPXEVT will set the event 29 flag of SER and no further VCP execution will occur until it is cleared.

10.1.2.3 Decisions Transfer

EDMA3 transfers from the decision buffer are VCPREVT frame-synchronized transfers. The programming of these transfers depend on the decision type and the traceback mode.

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

1. The decisions EDMA3 transfer parameters of the next user channel, if there is one ready to be decoded and the OUTF bit is 0.
2. Null EDMA3 transfer parameters (with all zeros), if there are no more user channels ready to be decoded and the OUTF bit is 0.
3. The output parameters EDMA3 transfer parameters, if the OUTF bit is 1.

10.1.2.4 Hard-Decisions Mode

The OPTIONS should be set as:

- 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)
- WIMODE = 0 (Normal operation)
- 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 transfer, each event triggers the transfer of ACNT elements)
- DAM = 0 (Dst address within an array increments. Dst is not a FIFO.)
- SAM = 1 (Src Address is fixed. Src is a FIFO.)
- SOURCE ADDRESS: VCPRDECS Decision FIFO address
- ACNT = (SYMR+1) × 8 (Number of hard decision bytes in an array)
- BCNT = CEIL(TNHD/ACNT) (Number of arrays in a frame) Where TNHD is the total number of hard decisions in bytes (Framelength/8).
- Destination Address: hard-decision array address
- SRCBIDX = 0
- DSTBIDX = ACNT
- SRCCIDX = 0

- DSTCIDX = 0
- CCNT = 1 (Number of frames in a block)

10.1.2.5 Soft-Decisions Mode

The OPTIONS should be set as:

- 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)
- WIMODE = 0 (Normal operation)
- 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 transfer, each event triggers the transfer of ACNT elements)
- DAM = 0 (Dst address within an array increments. Dst is not a FIFO.)
- SAM = 1 (Src Address is fixed. Src is a FIFO.)
- SOURCE ADDRESS: VCPRDECS Decision FIFO address
- ACNT = (SYMR+1) × 8 (Number of soft decision bytes in an array)
- BCNT = CEIL(TNSD/ACNT) (Number of arrays in a frame) Where TNSD is the total number of soft decisions (framelength)
- Destination Address: soft-decision array address
- SRCBIDX = 0
- DSTBIDX = ACNT
- SRCCIDX = 0
- DSTCIDX = 0
- CCNT = 1 (Number of frames in a block)

10.1.2.6 Output Parameters Transfer

This transfer is optional and depends on the OUTF bit. It is a 2- to 32-bit word VCPREVT frame synchronized transfer. The OPTIONS should be set as:

- 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)
- WIMODE = 0 (Normal operation)
- 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 transfer, each event triggers the transfer of ACNT elements)
- DAM = 0 (Dst address within an array increments. Dst is not a FIFO.)
- SAM = 0 (Src Address within an array increments. Src is not a FIFO.)
- SOURCE ADDRESS: VCP2 output register 0 address
- ACNT = 8
- BCNT = 1
- Destination Address: output register store array address
- SRCBIDX = 0
- DSTBIDX = 0

- SRCCIDX = 0
- DSTCIDX = 0
- CCNT = 1 (Number of frames in a block)

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

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

10.2 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). The values in each POLY*n* bit field must be entered in reverse order. The POLY*n* least-significant bit is set by the VCP2 logic.

- For rate 1/2, POLY0 and POLY1 are required. POLY2 and POLY3 must be set to zero.
- For rate 1/3, POLY0, POLY1, and POLY2 are required. POLY3 must be set to zero.
- For rate 1/4, all the POLY*n* bits are required.

The YAMT and YAMEN bits in VCPIC1 are described in Section 9.2.

The F and R bits in VCPIC2, the C bit in VCPIC3, and the TB bits in VCPIC5 are described in Section 9.1.

The IMAXI bits in VCPIC5 determine 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 13-bit signed values.

The SYMX and SYMR bits in VCPIC5 are described in Section 9.3 and Section 9.4.

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 EDMA3 programming (see Section 10.1.2.3).

11 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 9.2.

12 Event Generation

12.1 VCPXEVT Generation

A VCP2 transmit event (VCPXEVT) is generated when any of the following conditions appear:

- A START command write in VCPEXE.
- All input control words have been received and are correct.
- Top side or bottom side of the input FIFO buffer is empty.
- After all decisions have been read and OUTF is cleared, or if all decisions have been read and the output registers have been read and OUTF is set. Note that this extra XEVT pulls the next set of input configurations via EDMA3. If EDMA3 is not set up to link in another set of input configurations, then a dummy transfer should be set up to avoid an SER event flag being set for the EDMA3 parameter entry. If the flag is set, it effectively locks the VCP2, and must be cleared before any future events can be processed for that entry.

12.2 VCPREVT Generation

A VCP2 receive event (VCPREVT) is generated when any of the following conditions appear:

- The traceback unit has finished writing top side or bottom side of the output FIFO buffer.
- After the traceback is completed (the whole frame has been decoded).
- OUTF bit in VCPIC5 is 1 and all decisions have been read to read the output registers.

13 Operational Modes

- 0: 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 - CPU orders the coprocessor to start a processing block. The first action of the coprocessor is then to generate the first XEVT to trigger DMA transfer of the input control words.
- 2: Pause - CPU orders the coprocessor to pause a processing block at the beginning of traceback.
- 3: Unpause (single_traceback) - CPU orders the coprocessor to restart at the beginning of traceback and halt at next traceback.
- 4: Unpause (finish_traceback) - CPU orders the coprocessor to restart at the beginning of traceback and complete decode.
- Stop - CPU orders the coprocessor to reset. The coprocessor resets all VCP2 registers.

13.1 Debugging Features

Visibility into the internal operation of the VCP2 (i.e., the state metric accumulation, traceback memory) is available to the CPU via a pause command. However, since the *pause* command is not synchronized with the internal VCP2 state machine but is rather sent from the CPU at a random moment in time, this feature is of limited use.

The *pause* command on the VCP2 is augmented to provide visibility into VCP2 operation on a sliding window basis. Instead of using the normal *start* command which tells the VCP2 to perform a complete decode of one frame (including input/output transfers via EDMA3), *halt at beginning of traceback* and *resume until next traceback* commands are used, and the internal VCP2 memories can be inspected at various points in the decoding process. The procedure for using this command is as follows:

- VCP2 configuration and branch metrics are prepared
- A *halt at beginning of traceback* command is sent
- The VCP2 generates necessary interrupts to the EDMA3 to transfer input configuration and to start transferring branch metrics. The VCP2 performs state metric accumulation as branch metrics become available. When it reaches the end of the first sliding window (i.e., the reliability portion and the convergence portion), the VCP2 halts.
- The CPU polls the VCP2 status register until the VCP2 state changes from running to paused. At that point, the state metrics memory can be inspected, as well as the traceback memory. To perform an inspection, halt the CPU via a software breakpoint set at an appropriate point in the code, for instance. Then, the memory can be inspected visually via the debugger GUI, or, alternatively, the CPU can copy the relevant internal VCP2 memories to another location for later analysis.
  - The CPU sends the *resume until next traceback* command to the VCP2.
  - The VCP2 performs the traceback, generates a portion of hard or soft decisions, and continues with state metric accumulation until the end of the next sliding window (i.e., another number of R stages, where R is the reliability length).
  - The process continues until the decoding is complete. Alternatively, the decoding process can be run to completion after any sliding window by sending the *resume to completion* command instead of the *resume until next traceback* command.

14 Errors and Status

When the coprocessor detects an error, the coprocessor sets the status and error words, then sends an interrupt to the CPU. Any coprocessor processing is paused and the DSP must reset or start the coprocessor. An error occurs if the VCP2 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, the VCP2_INT interrupt is generated, and no processing is engaged. The only way to restart the VCP2 is to read VCPERR and send another START command. VCP2_INT has an interrupt selector value of 32. For details on how to set up interrupts, see the *TMS320C6000 DSP Interrupt Selector Reference Guide* (SPRU646).

The status registers are provided for debugging purposes and are best used when either the processor is halted or the VCP2 is halted. If an error occurs, the VCP2 is halted and a VCP2_INT 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 VCP2 is still running. One such case is when the VCP2 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 VCP2 configuration, in addition to some overhead to allow for EDMA3 usage.

Appendix A Revision History

This revision history highlights the technical changes made to the document in this revision.

Table 31. C645x Revision History

See	Additions/Modifications/Deletions
Table 11	Modified bits 15-0, FL, description
Section 9.1.4	Modified step 4 in numbered list Modified last paragraph

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

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