IEEE802-Compatible Viterbi Decoder v1.1

DS204 November 10, 2004 **Product Specification**

Powered by L3 communications

Features

- Viterbi Decoder, jointly developed by L-3 Communications and Xilinx, reaches OC3 (155Mbps) data rates and higher
- Available for all Virtex™, Virtex-E, Virtex-II™, Virtex-II Pro™, Virtex-4™, Spartan-II™, Spartan™-IIE, and Spartan-3™ FPGAs
- Industry-standard constraint length 7, (G0,G1)=(171,133) or (133,171) compatible with Q1900, DVB, IEEE802.11a, IEEE802.16a, HiperAccess, HiperMan, INTELSAT IESS-308/309
- Adaptive rate change via puncturing interface
- Best state logic in traceback reduces latency and improves Bit Error Rate (BER) up to 0.65dB at rate 7/8; necessary for TDMA burst modems using high code rates.
- Parameterizable options for soft data input, accumulated path metric, and best state widths
- Fully synchronous one-clock version (Viterbi1x, see Figure 2) for adaptive traceback length up to 126
- Fully synchronous 2x clock version (Viterbi2x, see Figure 2) for reduced latency, lower power dissipation, and half Block RAM size; with adaptive traceback length of 48 or 96
- The Viterbi2x version has a 29 clock latency for IEEE802.11a PLCP header 24-bit SIGNAL symbol using zero-tail termination control signals
- Normalization output for synchronization and indication of data link SNR
- Viterbi2x supports BER monitor
- Alternative branch metric inputs for Pragmatic Trellis Coded Modulation (PTCM)
- Erasure capability of framing/synchronization patterns
- VHDL source code or netlist distribution with VHDL channel model for BER simulation
- Incorporates Xilinx Smart-IP™ technology for maximum performance

LogiCORE™ Facts	
Core Specifics	
Supported Device Family	Virtex, Virtex-E, Virtex-II, Virtex-II Pro, Virtex-4, Spartan-II, Spartan-IIE, Spartan-3
Resources Used	See Table 9 and Table 9
Special Features	RPM Core
Provided with Core	
Documentation	Product Specification
Design File Formats	VHDL and EDIF
Constraints File	UCF
Verification	VHDL Testbench
Instantiation Template	VHDL Wrapper
Reference Designs & application notes	None
Additional Items	None
Design Tool Requirements	
Xilinx Implementation Tools	ISE 6.1.03i or later
Verification	ModelSim PE 5.4e
Simulation	ModelSim PE 5.4e
Synthesis	Synplify Pro 7.1
Support	
Support provided by Xilinx, Inc.	

NOTICE OF DISCLAIMER: Xilinx is providing this design, code, or information "as is." By providing the design, code, or information as one possible implementation of this feature, application, or standard, Xilinx makes no representation that this implementation is free from any claims of infringement. You are responsible for obtaining any rights you may require for your implementation. Xilinx expressly disclaims any warranty whatsoever with respect to the adequacy of the implementation, including but not limited to any warranties or representations that this implementation is free from claims of infringement and any implied warranties of merchantability or fitness for a particular purpose.

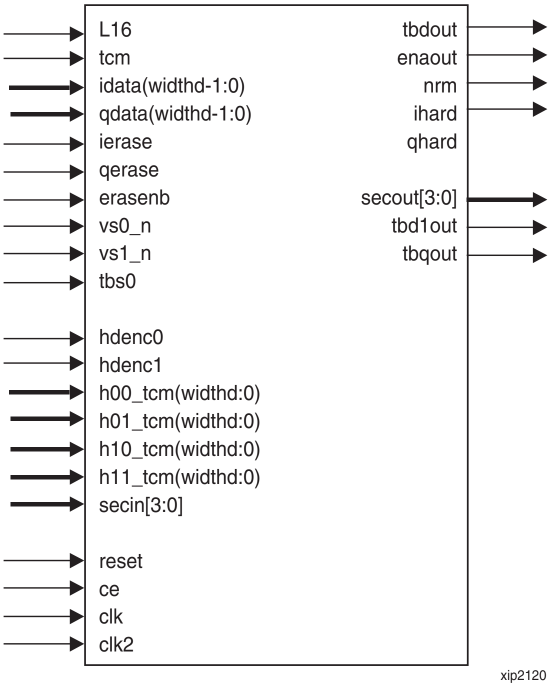


Figure 1: **Virtex2x Decoder VHDL Interface Signals**

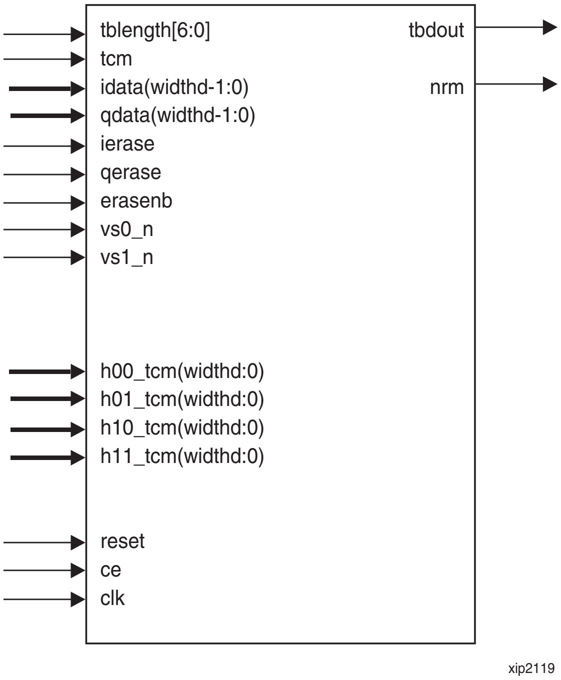


Figure 2: **Viterbi1x Decoder VHDL Interface Signals**

Functional Description

The core implements a Viterbi Decoder for decoding convolutionally encoded data. The basic architecture of a convolutional encoder is shown in Figure 6. In the convolutional encoder, the incoming data is brought into the constraint register a bit at a time. The two required output bits are then generated by modulo-2 addition of the required bits from the constraint register. The bits to be XORed are selected by the convolutional codes as shown in Figure 6. The Viterbi core decodes the data generated by the convolutional encoder that has a constraint length k=7 and the two generator polynomials G0=171 and G1=133 as shown in Figure 6. Simply swapping idata and qdata inputs allows the same Viterbi decoder core to decode data encoded with generator polynomials G0=133 and G1=171. The Viterbi core has a fully parallel architecture, which provides fast data throughput. The core also supports puncturing rates n/(n+1) through an external interface that generates the signals vs0_n and vs1_n. Puncturing gives a large range of transmission rates and reduces the bandwidth requirement on the channel. See the *Puncturing* section. The decoder core consists of three main blocks as shown in Figure 7.

Branch Metric Calculation

The first block is the Branch Metric Unit (BMU). This module costs the incoming data using a Euclidean metric. The incoming data is soft coded with a parameterizable bit width that can be set to any value. It is possible to bypass the BMU with external PTCM branch metrics. The data format of the Viterbi core is offset binary. The conversion between offset two's complement and offset binary is shown in Table 1. For offset two's complement, the analog signal midpoint is between 000 and 111. For offset binary, the analog signal midpoint is between 011 and 100. Even though the soft data representation is offset; this does not imply the analog input signal should have any DC offsets. Any analog signal DC offset before conversion to soft data inputs will result in reduced BER and higher normalization rates, and thus should be avoided.

Table 1: **Data Format for Soft Width 3**

	Offset Two's Complement	Offset Binary
Strongest 0	011	000
	010	001
	001	010
Weakest 0	000	011
Weakest 1	111	100
	110	101
	101	110
Strongest 1	100	111

Add Compare Select

The second block in the decoder is the Add Compare Select unit (ACS). This module selects the optimal path to each state in the Viterbi trellis (survival path). Figure 8 shows one stage in the Viterbi trellis for a constraint length 3 decoder. The ACS module decodes for each state in the trellis, thus for a constraint length 7 decoder that has 64 states there are 64 sub-blocks in the ACS block. See the section on Core Resource Utilization for further characterization of the decoder. The ACS has a Best State Calculation

(BSC) module which finds the state with the smallest accumulated path metric to start each traceback block.

The BSC module output, used in traceback, improves BER up to 0.65dB for rate 7/8 compared to Qualcomm Q1900 in continuous mode; improvement is greater for burst operation. Generic Viterbi decoders without a BSC module require longer traceback lengths to achieve equivalent BER performance. Increasing traceback lengths results in increased Viterbi decoder latency, which is unacceptable in many burst/TDMA applications. In applications with high code rates, the BSC module again is fundamental in reducing traceback length for any given BER and latency requirement. If latency and traceback length are not critical requirements, the BSC module can be removed for substantial size savings, as described in the core utilization section.

For applications like IEEE802.11a with zero-tail packet termination, the input control signal tbs0 allows the user to bypass BSC output for the last block in the packet and substitute zero state address instead of best state address. This combination of zero state for the last block and best state addresses for all other blocks in the packet reduces latency and provides higher BER performance for other blocks in the packet.

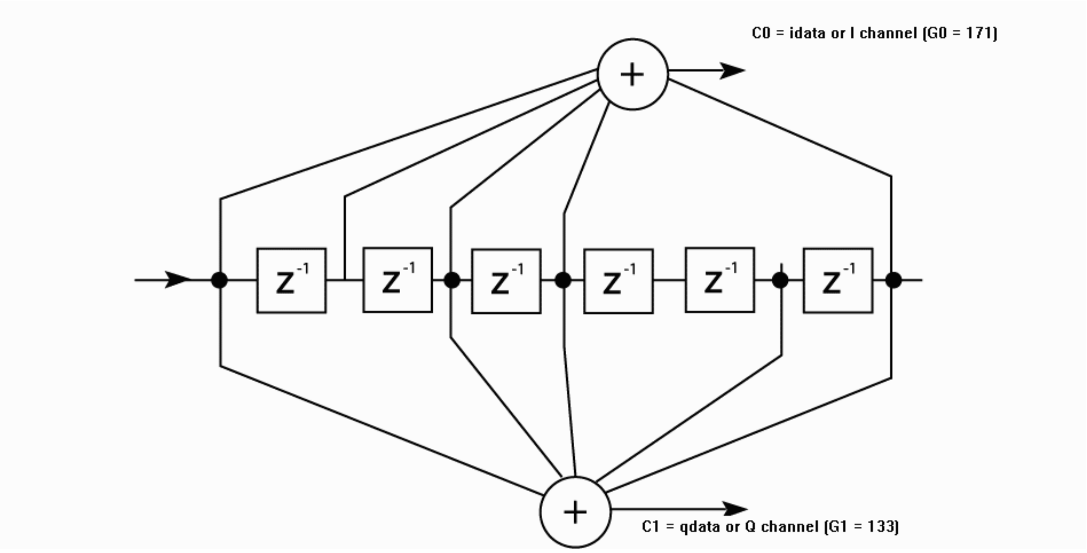

C0 = idata or I channel (G0 = 171)
+
Z-1
Z-1
Z-1
Z-1
Z-1
Z-1
+
C1 = qdata or Q channel (G1 = 133)

Figure 3: **Convolutional Encoder of Constraint Length 7**

Traceback

The third block in the decoder is the traceback module (see Figure 7). This module finds the optimal path through the Viterbi trellis by tracing through a fixed number of ACS results. For the two-clock version, the recommended traceback length is 48 for non-punctured codes and 96 for punctured codes. For the one-clock version, the traceback length can be up to 126 for Virtex-II but is limited to 62 for non Virtex-II families. The traceback block is implemented in block RAM. For additional details, see the Core Resource Utilization section.

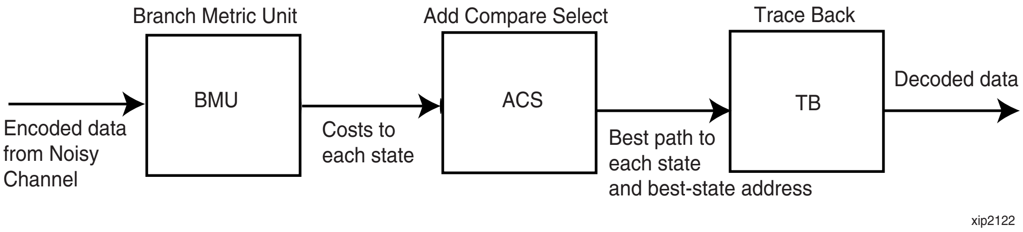


Figure 4: **Viterbi Decoder Block Diagram**

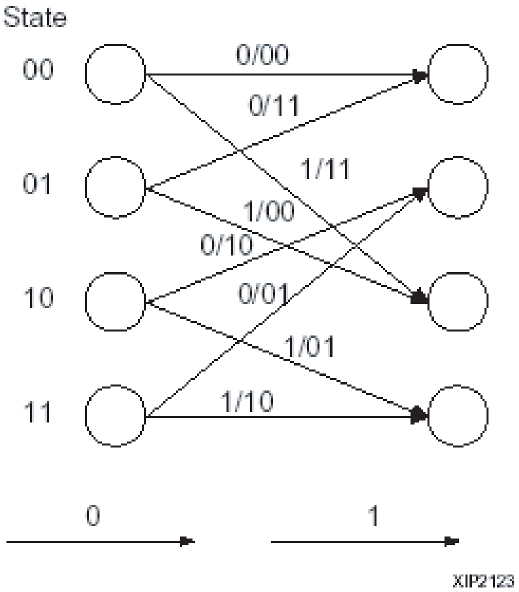


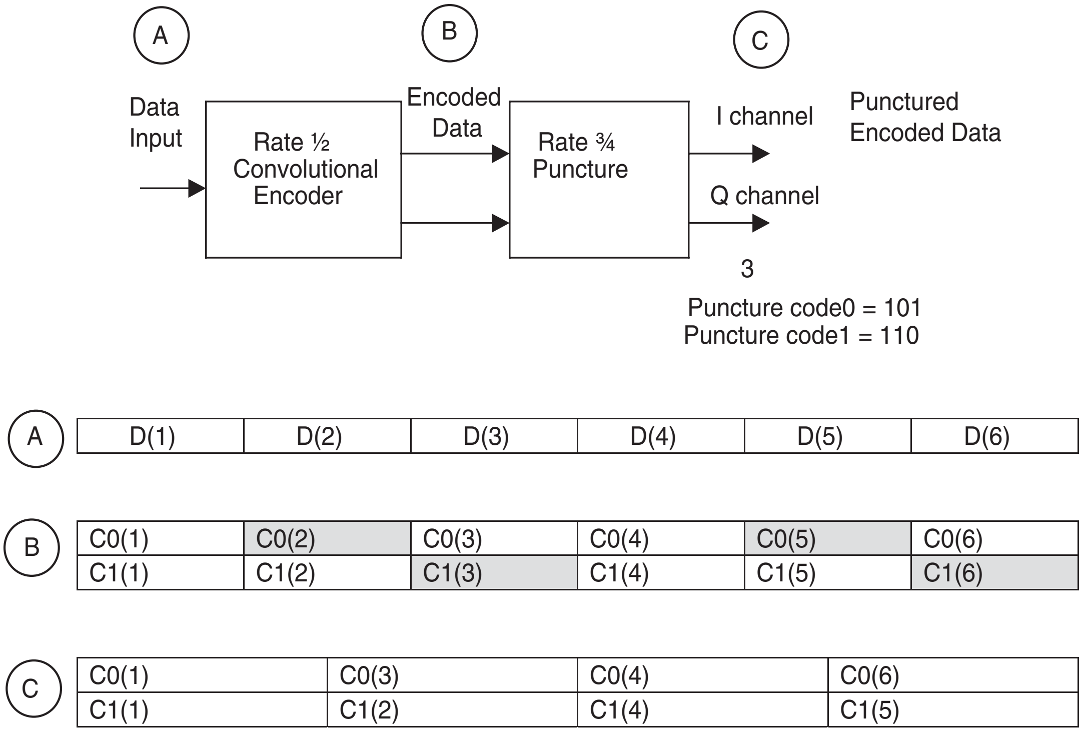
Figure 5: **State Transitions for Constraint Length 3**

Puncturing

Data can be punctured prior to transmission. The convolutional encoder is always a rate ½ encoder, 2 bits output for every 1 bit input. After encoding, certain bits of the rate ½ encoded data are punctured (or deleted) and not transmitted. Thus, for a rate ¾ punctured encoder, for every three input bits, only four out of the six encoded bits generated by the encoder are actually transmitted (see Figure 9).

The encoder output is punctured according to a pair of puncture codes as shown in Table 2. The puncture codes are a bit pattern that identifies which bits from the encoder are to be transmitted. The use of puncturing significantly reduces the number of bits to be transmitted over the channel. For a puncture code of length n, exactly (n+1) bits will be transmitted; *i.e.*, the rate of the encoder will be n/(n+1). For rate n/(n+1) puncturing, the length of each puncture code must be n bits and the total number of non-zero bits in all the puncture codes must be equal to (n+1).

The decoder supports decoding of punctured data through an external interface. This external interface depunctures the punctured data prior to the Viterbi decoding and generates the input signals to the Viterbi decoder vs0_n and vs1_n (see Figure 7). Depuncturing can be done by inserting any value in the punctured location. For non-punctured codes, vs0_n and vs1_n are both High. For punctured codes, vs0_n and vs1_n will have the pattern shown in Figure 7.



xip2124

Figure 6: **Puncturing Encoded Data with 3/4 Puncture Rate and Quadrature-Channel Output**

Table 2: **Example Puncture Codes for Various Input Rates**

Rate	Puncture Code 0	Puncture Code 1
2/3	10	11
3/4	101	110
4/5	1000	1111
5/6	10101	11010
6/7	100101	111010
7/8	1000101	1111010

BER Monitor

The two-clock version supports BER through an external circuit shown in Figure 8. The decoder delays the channel hard decision data and aligns them with the Viterbi decisions. Decoded data from the Viterbi decoder are re-encoded using the convolutional encoder and compared with the delayed channel hard decisions. An error is indicated if the delayed and encoded data differ. The two sets of symbols can differ if there is an error on the channel or if the Viterbi decoder has decoded incorrectly. The probability of the decoder incorrectly decoding is significantly smaller than the probability of a channel bit error. Therefore, the BER output gives an accurate measurement of the errors on the channel, given that the Viterbi decoder is already synchronized. For the one-clock version, the hard decision delay line must be provided in the external circuit.

Normalization

The Viterbi core provides a normalize output signal. Normalize is asserted when all the path state variables have reached half their maximum value; then, all state variables are reduced by that amount to avoid overflow. The normalize signal can be used to detect Viterbi synchronization. A high normalization rate (exceeding a certain threshold) indicates loss of synchronization. A low normalization rate (less than a certain threshold) indicates Viterbi synchronization has been achieved. In general, the normalization rate is inversely proportional to the Signal to Noise Ratio (SNR) at the decoder input. For an in-depth discussion on low SNR (<2dB) synchronization using normalization signal in conjunction with BER monitor, see the *Viterbi Synchronization* document available from the Xilinx DSP marketing team.

Trellis Mode

Coding is used for power-limited communication systems, while higher-order modulation schemes are used for bandwidth- limited systems. Pragmatic trellis coded modulation (PTCM) uses coding combined with higher-order modulation schemes to get a compromise between power efficiency and bandwidth. PTCM transmits coded and uncoded data in the same I and Q symbols. The Viterbi core supports decoding of PTCM systems. The four PTCM branch metrics and the sector number can be computed from the received I and Q symbols through a look up table. The Viterbi core processes the four PTCM branch metrics to decode the coded data bits and delays the sector number by the Viterbi latency. The uncoded bits can be decoded using the Viterbi decisions and the delayed sector numbers

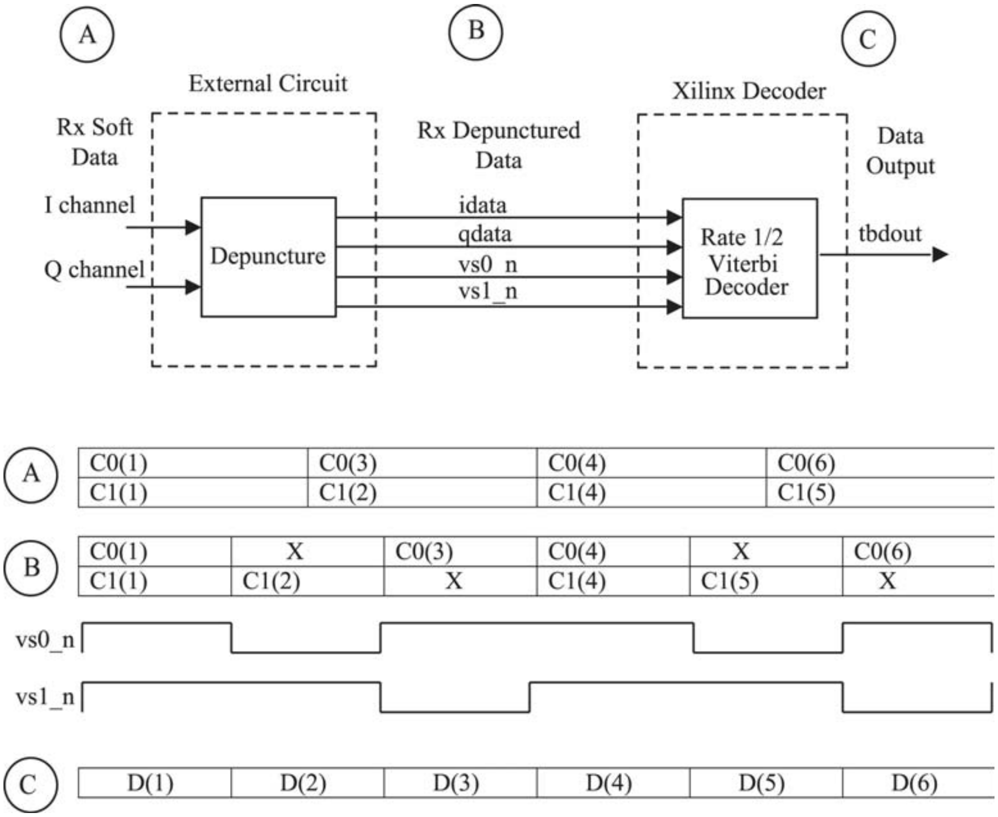


Figure 7: **Depuncturing Rate 3/4 Punctured Data with Quadrature-Channel Soft Input**

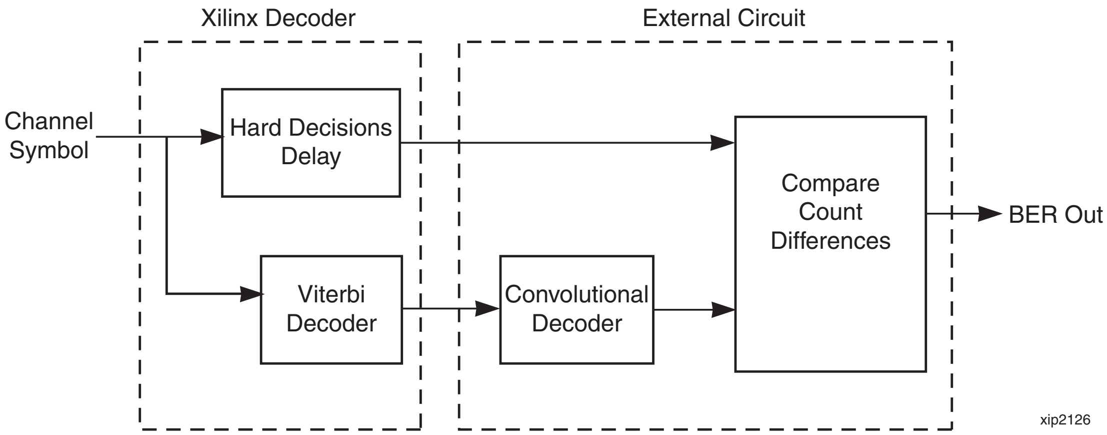


Figure 8: **Bit Error Rate Calculation**

Pinout

The symbols for the two-clock and the one-clock versions of the Viterbi core are shown in Figure 1 and Figure 2 respectively. Table 3 and Table 4 describe the core signals for the versions. Timing diagrams for the signals are shown in *"Clocking Schemes" on page 14*.

Table 3: **Viterbi2x Core Signals**

Name	Direction	Description
L16	Input	When asserted, traceback length = 48; otherwise, traceback length = 96
tcm	Input	When asserted, enables the PTCM mode
idata	Input	I channel soft data input
qdata	Input	Q channel soft data input
ierase	Input	I channel erasure if erasenb is asserted
qerase	Input	Q channel erasure if erasenb is asserted
erasenb	Input	When asserted, enables the erasure capability of the viterbi
vs0_n	Input	When asserted (Low), puncture I data; ignore idata
vs1_n	Input	When asserted (Low), puncture Q data; ignore qdata
tbs0	Input	When asserted, traceback starts from zero state; otherwise, traceback starts from best state
hdenc0	Input	I channel hard decision values in PTCM mode
hdenc1	Input	Q channel hard decision values in PTCM mode
h00_tcm	Input	The first branch metric (H00) in PTCM mode
h01_tcm	Input	The second branch metric (H01) in PTCM mode
h10_tcm	Input	The third branch metric (H10) in PTCM mode
h11_tcm	Input	The fourth branch metric (H11) in PTCM mode
secin	Input	Sector number in PTCM mode
reset	Input	Synchronous reset: reinitializes core logic
ce	Input	Clock Enable: i.e., Data input enable
clk	Input	1x clock: Data input clock, clk and clk2 must be rising edge aligned
clk2	Input	2x clock used in traceback trellis read operation; clk and clk2 must be rising edge aligned
nrm	Output	Normalize signal
tbdout	Output	Traceback decoded output data (Viterbi decisions)
enaout	Output	Output data enable
ihard	Output	I channel hard decision data (or hdenc0 in PTCM mode) delayed by decoding latency
qhard	Output	Q channel hard decision data (or hdenc1 in PTCM mode) delayed by decoding latency

Table 3: **Viterbi2x Core Signals** ***(Continued)***

Name	Direction	Description
secout	Output	Secin delayed by decoding latency
tbd1out	Output	Reverse order traceback output data at 2x clock rate
tbqout	Output	Indicates start of traceback; upon deassertion, traceback data for two blocks is output on signal tdb1out

Table 4: **Viterbi1x Core Signals**

Name	Direction	Description
tblength	Input	Traceback length
tcm	Input	When asserted, enables the PTCM mode
idata	Input	I channel soft data input
qdata	Input	Q channel soft data input
ierase	Input	I channel erasure if erasenb is asserted
qerase	Input	Q channel erasure if erasenb is asserted
erasenb	Input	When asserted, enables the erasure capability of the viterbi
vs0_n	Input	When asserted (Low), puncture I data; ignore idata
vs1_n	Input	When asserted (Low), puncture Q data; ignore qdata
h00_tcm	Input	The first branch metric (H00) in PTCM mode
h01_tcm	Input	The second branch metric (H01) in PTCM mode
h10_tcm	Input	The third branch metric (H10) in PTCM mode
h11_tcm	Input	The fourth branch metric (H11) in PTCM mode
reset	Input	Synchronous reset: reinitializes core logic
ce	Input	Clock Enable: i.e., Data input enable
clk	Input	1x clock: Data input clock
tbdout	Output	Traceback decoded output data (Viterbi decisions)
nrm	Output	Normalize signal

idata, and qdata Inputs

These are the soft data inputs to be decoded. The generic widthd specifies the number of data bits. The format for 3-bit soft data case is shown in Table 1. The recommended generic values are shown in Table 7.

ierase, qerase, and erasenb Inputs

These signals enable the erasure capability of the Viterbi core. The erasure capability is used when the error location is known. An example would be the erasure of the synchronization pattern overlay for the Reed Solomon decoder framing in a concatenated FEC combination of Viterbi and Reed-Solomon decoders.

vs0_n and vs1_n Inputs

These input signals are used for decoding punctured codes. For nonpunctured codes, vs0_n and vs1_n are both de-asserted. For rate n / (n+1) punctured codes, vs0_n and vs1_ will have the pattern shown in Table 2.

hdenc0 and hdenc1 Inputs

The I and Q channel hard decision values in PTCM mode. The hdenc0 and hdenc1 signals are delayed inside the core by the decoding latency for BER monitoring purposes in PTCM mode.

h00_tcm, h01_tcm, h10_tcm, and h11_tcm Inputs

These signals are the alternate four branch metrics used in PTCM mode. These metrics can be computed external to the core through a look up table. The four branch metrics are generic of widthd+1; *i.e.*, the number of bits in the branch metrics is equal to the number of bits in the soft data input plus one. These inputs should conform to offset-binary format shown in Table 1.

secin Input

This signal is the sector number used in PTCM mode. The sector number can be computed outside the core through an external circuit. The core delays the sector numbers by the decoding latency. The delayed sector numbers with the Viterbi decisions can be used to decode the transmitted uncoded bits in PTCM systems.

reset Input

When reset is asserted (High), all the core flip-flops are synchronously initialized. The core will remain in this state until it is deasserted. If clk and clk2 are generated using a DLL / DCM (Digital Clock Manager), the reset input signal must be asserted after the DLL / DCM is locked.

ce Input

When ce (clock enable) is High, data is clocked into the core at the rising edge of the 1x clock (clk).

clk and clk2 Inputs

Data input are clocked into the core at the rising edge of clk. The BMU, ACS and the write part of the traceback are connected to clk. Only the read part of the traceback uses the 2x clock (clk2). Clk and clk2 shall be rising edge aligned within the tolerance of a DLL or DCM clock generator. It is possible to source clk with the same clock source as clk2 if the ce signal has a duty cycle of 50 percent or less. If either clk or clk2 lose lock, the Viterbi core shall be reset after clocks relock.

tbdout Output

This is the decoded data output.

enaout

The output data enable signal indicates the valid data on the tbdout output. Enaout is equivalent to the input signal ce delayed by decoding latency.

nrm Output

The nrm output signal indicates when normalization has occurred within the core, thus giving an immediate indication of the rate of errors in the channel. Normalization also can be used to synchronize the Viterbi core. See the section on "Normalization" earlier in this data sheet.

ihard and qhard Outputs

When tcm is asserted, ihard and qhard are hdenc0 and hdenc1 delayed by decoding latency. When tcm is deasserted (Low), ihard and qhard are the hard decision values of idata and qdata delayed by decoding latency. Signals ihard and qhard can be compared externally to the Viterbi decisions for BER monitor purposes.

secout Output

Signal secout is secin delayed by the decoding latency. It is used with the re-encoded Viterbi decisions to decode the transmitted uncoded bits in PTCM systems.

tbs0 Input and tbd1out, tbqout Outputs

These signals are used in the Viterbi2x core to achieve lower latency when using zero-tail packet termination applications like IEEE802.11a Viterbi decoder. When tbs0 is asserted for the last 16 idata/qdata symbol pairs of the zero tail block at the end of a packet; then traceback uses zero state address to start the traceback of this last block. All other blocks in the packet use best-state address for their traceback starting address.

For zero-tail termination, the output signals tbd1out and tbqout provide access to the decode data of the last two blocks. When tbqout is asserted for two 2x clock periods, then the last two decoded data blocks streams out at the 2x clock rate on the signal tbd1out independent of the clock enable (ce) input control signal. Output data on tbd1out bypasses the internal LIFOs of the core to reduce latency; thus the data streaming on tbd1out is in reverse order.

Core Generic Parameters

The Viterbi core has generic parameters that are initialized inside the VHDL entity declaration. The generic values should be changed only in the top-level entity (viterbi.vhd). The tools will pass those values to the lower-level entities. Table 5 and Table 6 describe the core generic parameters. Table 7 lists the recommended values for the generic parameters.

Table 5: **Viterbi2x Core Generic Parameters**

Generic Parameter	Description
Target	Target family (Virtex or Virtex-II)
Spartan3	1 when targeting Spartan-3 or Virtex-4; 0 when targeting other devices
V2muxf	0 when targeting Virtex and 1 when targeting Virtex-II
V2bram	0 when targeting Virtex and 1 when targeting Virtex-II
widthd	Number of soft data input bits
widtha	Number of bits used in the path metric calculation
widths	Number of bits used in the best state calculation
rloc_xorigin	Sets the top-level X coordinate for the core CLBs

Table 5: **Viterbi2x Core Generic Parameters *(Continued)***

Generic Parameter	Description
rloc_yorigin	Sets the top-level Y coordinate for the core CLBs
ramb_xorigin	Sets the X coordinate for the core BRAMs
ramb_yorigin	Sets the Y coordinate for the core BRAMs

Table 6: **Viterbi1x Core Generic Parameters**

Generic Parameter	Description
Target	Target family (Virtex or Virtex-II)
Spartan3	1 when targeting Spartan-3 or Virtex-4; 0 when targeting other devices
V2dram	0 when targeting Virtex and 1 when targeting Virtex-II
V2bram	0 when targeting Virtex and 1 when targeting Virtex-II
widthd	Number of soft data input bits
widtha	Number of bits used in the path metric calculation
widths	Number of bits used in the best state calculation
rloc_xorigin	Sets the top-level X coordinate for the core CLBs
rloc_yorigin	Sets the top-level Y coordinate for the core CLBs
ramb_xorigin	Sets the X coordinate for the core BRAMs
ramb_yorigin	Sets the Y coordinate for the core BRAMs

Target

This parameter refers to the target FPGA family, either Virtex or Virtex-II. When Virtex is chosen, the core is targeting the Spartan-II, Spartan-IIE, Virtex, and Virtex-E families. When Virtex-II is chosen, the core is targeting the Virtex-II and Virtex-II Pro families.

Spartan3

When targeting Spartan-3 or Virtex-4 the generic parameter Spartan3 is set to 1, while the remaining generic parameters are set to their Virtex-II values.

V2muxf

V2muxf is a generic value that is 0 when targeting the Virtex family and 1 when targeting the Virtex-II family. It changes the way that MUXF5 and MUXF6 are floor planned.

V2bram

V2bram is a generic value that is 0 when targeting the Virtex family and 1 when targeting the Virtex-II family. When it is 0, the number of Block RAMs used in the traceback is doubled.

V2dram

V2dram is a generic value that is 0 when targeting the Virtex family and 1 when targeting the Virtex-II family. It changes the way the distributed RAMs are floor planned.

widthd

This parameter indicates the number of soft I and Q data input bits. The number of bits used in the branch metric calculation is always equal to widthd+1, which is also equal to the number of bits of the branch metrics h00_tcm, h01_tcm, h10_tcm and h11_tcm in PTCM mode.

widtha

This parameter indicates the number of bits used in the path metric calculation. It depends on “widthd” shown in Figure 7.

widths

This parameter indicates the number of bits used in the best state calculation, widths <= (widtha-3). Only the MSBs of the accumulated path metrics are used in best state calculation. For the 3-bit soft data input, the 3 MSBs of the path metrics are used in best state calculation. Table 7 shows the possible generic parameter options for widthd, widtha, and widths.

Table 7: **Viterbi Generic Parameter Selection**

widthd	widtha	widths
3	6	3
4	8	3,4,5
5	8	3,4,5
6	10	4..7
7	10	4..7
8	12	5..9

rloc_xorigin, rloc_yorigin, ramb_xorigin, and ramb_yorigin

When targeting Virtex-II, the default coordinates start from the left bottom corner of the FPGA at (x,y)=(0,0), while when targeting the Virtex and Virtex-E, the default coordinates start from the left top corner of the FPGA at (1,1) for the CLBs and at (0,0) for the BRAMs.

To achieve highest performance, these coordinates must be chosen as follows: Choose the generic parameters “ramb_xorigin” and “ramb_yorigin” to be the X and Y coordinates of any BRAM in the FPGA (you can look up those coordinates using the FPGA editor), then the generic parameters “rloc_xorigin” and “rloc_yorigin” will be the X and Y coordinates of the CLB to the right of that BRAM. Make sure the coordinates you choose for the BRAM and the CLB do not cause the design to exceed the dimensions of the FPGA. It is a good practice to implement the design first, using the default coordinates, then try to move the design to the desired place inside the FPGA by changing the generic values of the X and Y coordinates.

Clocking Schemes

The Viterbi2x core has two modes of operation.

Mode 1

In the first mode of operation, clk2 is used in the read part of the traceback section and yields the highest data rate capability of the Viterbi2x core. This is the normal mode of operation and it requires two clocks: clk and clk2. These clocks, clk and clk2, are rising edge aligned to within the tolerance of a DLL / DCM. After clk and clk2 are stable and locked, the reset input signal is asserted for at least one clock period before input data is processed by the core. If the DCM / DLL loses lock, the reset process is repeated. The highest data rate mode of operation is shown in Figure 9. It is also possible to reduce traceback latency by using clock rates higher than input data rate. In this case, the clock enable (ce) signal controls periodic or random data inputs with respect to 1x clock (clk). The data output enable signal (*enaout*) indicates when new decoded data is available based on the input clock enable.

Mode 2

In the second mode of operation, the Viterbi2x core uses a single 2x clock for both *clk* and *clk2* input signals. In this mode, the data rate shall be 50 percent or less of the *clk2* rate, and shall be controlled via the clock enable (*ce*) signal. Figure 10 shows a 50 percent duty cycle for the *ce* signal, but it is possible to operate at any periodic or random input data rate of 50 percent or below duty cycle. Traceback latency is further reduced for higher clock rate versus data rate ratios. The Viterbi1x core has a single clock input. Its highest data rate mode of operation is shown in Figure 11. If clock rate is higher than the input data rate, the *ce* signal controls the flow of slower input data into the core.

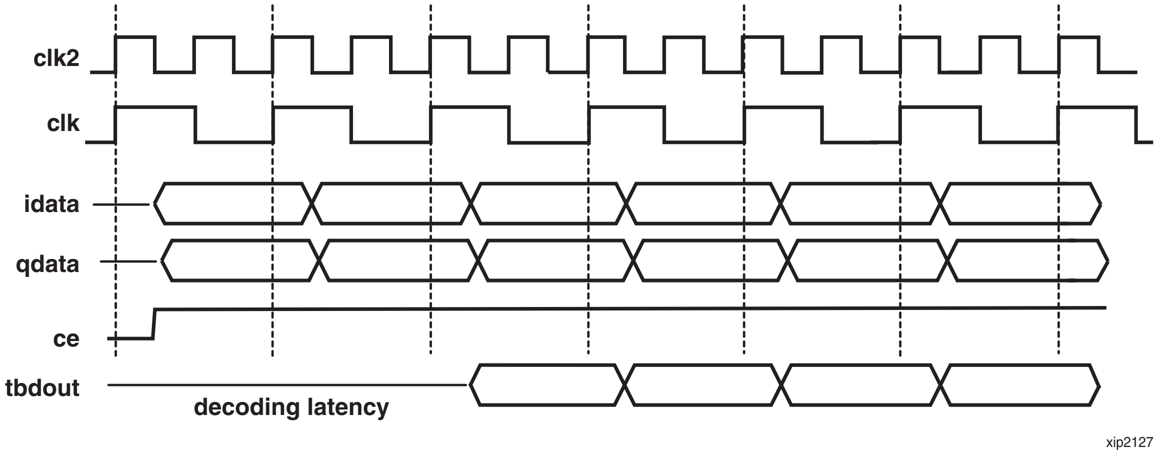


Figure 9: **Viterbi2x Normal Mode of Operation Timing Diagram**

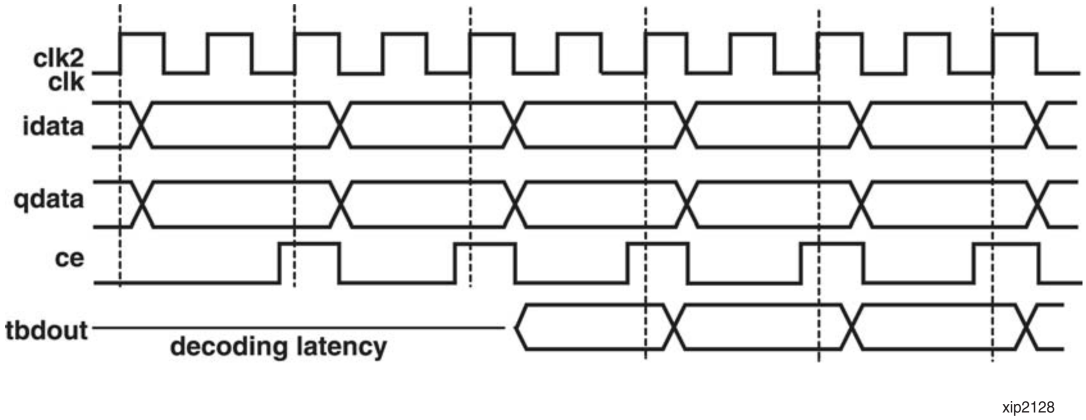


Figure 10: **Viterbi2x Timing Using One Clock for Both Domains**

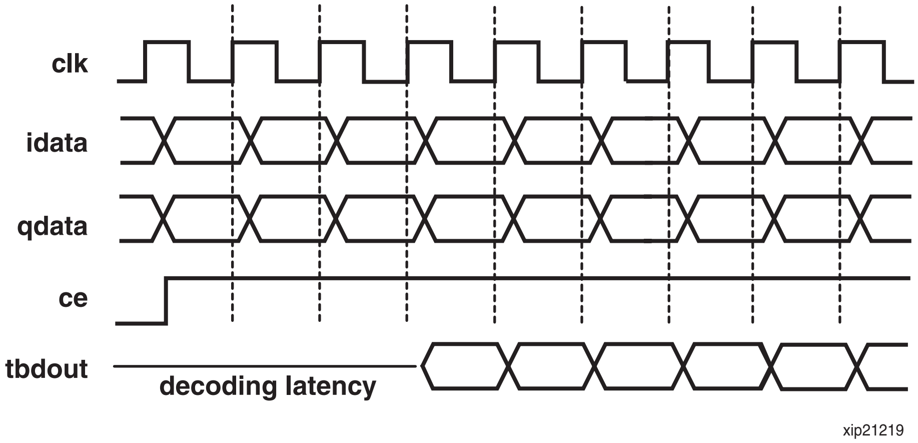


Figure 11: **Viterbi1x Timing Diagram**

Latency

The latency of the core depends on the traceback length and input data rate. Latency is a count of the number of ce inputs between data input and data output. For Viterbi2x, latency is 18 plus 3x traceback length. If data rate is reduced, latency will be less than 3x traceback length and approaches 2x traceback depth plus 15. Table 8 shows the latency of Viterbi2x core for different traceback length and data input rate. For Viterbi1x, latency is 24 plus 4x traceback length. Viterbi2x with full best state pipeline requires another two clocks over standard Viterbi2x latency.

The latency for terminated blocks is lower than continuous data latency described in the previous paragraph and is listed in Table 10. For tail-biting terminated blocks the latency is 17 plus 2x traceback length. For zero-tail terminated blocks the latency is 17 plus 1x traceback depth. These reduced latencies are achieved though use of the special terminated block signals tbs0, tbd1out, and tbqout.

Table 8: **Virterbi2x Core Latency**

Data Rate	1x Input Clock	1/2 Input Clock	1/4 Input Clock	1/8 Input Clock
Traceback Length 48	162	136	123	117
Traceback Length 96	306	256	231	219

IEEE802.11a Recommendations

During the "PLCP training Preamble", the user has time to input (96+(96-24)) or 168 strong zeros (or soft zeros corresponding to link SNR) into the Viterbi802 decoder which is essentially latency free. The Viterbi trellis is forced to the zero state and at the same time primes the decoder to be within 24 bits from starting traceback. After the FFT processes the SIGNAL OFDM symbol and inputs the 24 I/Q pairs into the Viterbi decoder with tbs0 control signal asserted for zero-tail termination; then the decoder asserts tbqout, starts traceback, and the decoded data immediately streams out in reverse order on tbd1out. After the 24 SIGNAL OFDM data bits are output, the rest of the 168 bits are flushed. The latency in determining the following DATA field code rate from the SIGNAL data bits is reduced to 29 clocks. During padded zero flushing of the SIGNAL symbol, the DATA field I/Q pairs are input into the decoder with appropriate code rate (de-puncturing pattern). Further decoding follows traditional Viterbi sliding window processing with best state traceback address until the end of the DATA field packet is reached. The same zero-tail termination approach is used at the end of the DATA field set of OFDM symbols to reduce the latency at the end of the packet from 306 clocks down to 113 clocks.

RPM (Relationally Placed Macro)

The Viterbi decoder is built with relative location attributes attached. When a core is created as an RPM, it is possible that one or more of the core dimensions may exceed those of the device being targeted. If this is the case, mapping errors will occur and the compilation process will fail. In this case, one can remove the RLOC attributes by editing the generated EDIF file. However, core performance will no longer be guaranteed.

Core Resource Utilization

The area of the core increases with the soft width of the input data. Some example configurations are shown in Table 10. If the additional 0.2dB to 0.65dB BER improvement gained by the BSC is not required, the BSC can be removed. Eliminating the BSC function reduces the core slice count by more than 20 percent and reduces the number of flip flops in the remaining slices by 10 percent.

Block RAM Utilization

Because of the larger blocks of RAM available in the Virtex-II family of devices, the Block RAM requirements will vary depending on whether the core is implemented in Spartan-II, Spartan-IIE, Virtex, Virtex-E, or in Virtex-II or Virtex-II Pro. See Table 9.

Table 9: **Block RAM Requirements for the Viterbi Cores**

Design	Virtex, Virtex-II, Spartan-II, and Spartan-IIE	Virtex-II, Virtex-II Pro, Virtex-4, and Spartan-3
Viterbi2x	4	2
Viterbi1x	8	4

Performance Characteristics

It is important to set a reasonable period constraint on the core input clocks. Over constraining the clock period will result in long route time and may even compromise the speed performance. The data shown in Table 10 are obtained when targeting the Virtex-II XC2V1000 FG256-6 device, with version 1.96 speed files and 4.1i PAR tools. The results may change slightly by trying different speed constraints. If necessary, performance can be increased by selecting a part with a faster speed grade.

When targeting the XC2VP20-7 using the ISE 6.3.01i PAR tools with Soft Data Width value of 3, the Viterbi2x runs at 186 MHz and has 1085 slices, while the Viterbi1x runs at 186 MHz and has 1083 slices. When targeting XC4VLX25-11 using the ISE 6.3.01i PAR tools with Soft Data Width value of 3, the Viterbi2x runs at 192 MHz and has 1085 slices, while the Viterbi1x runs at 200 MHz and has 1085 slices.

For Spartan-3 devices only, when targeting the XC3S1000-4 using the ISE v6.1.03i PAR tools at a Soft Data Width value of 3, the Viterbi2x runs at 102 MHz and has 1085 slices; while the Viterbi1x runs at 115 MHz and has 1083 slices.

Table 10: **Viterbi-II/Viterbi Core Implementation Results for Widths 3, 4, and 5**

Design	Viterbi2x				Viterbi1x	
Traceback Length	48	96	48	96	48	126
Soft Data Width	3	3	4 or 5	4 or 5	3	4 or 5
Area (Slices)	1084	1084	1346	1346	1076	1338
Area/No BSC	897	897	1159	1159	889	1151
Block RAM	2	2	2	2	4	4
Continuous Latency	162	306	162	306	216	528
Tail-Bite Latency	113	209	113	209	168	402
Zero-Tail Latency	65	113	65	113	N/A	N/A
Maximum Clock Frequency (MHz)	157	157	154	154	156	154

Hardware Verification

The Viterbi802 decoder was tested in the lab using a Virtex-II FG676 Proto Board populated with an XC2V3000-6 FG676 device. A hardware testbench was developed to test the BER performance of the Viterbi802 decoder in an additive white Gaussian noise (AWGN) channel. The testbench consists of an LFSR-based data generator, the convolution encoder, the Xilinx AWGN Core, an adder, the Viterbi802 decoder core, a bit counter, an error counter and other miscellaneous support circuitry.

The LFSR-based data generator generates a random binary (0,1) sequence, which is convolutionally encoded. The encoded data is mapped to I/Q symbols in the two-dimensional Euclidean space (i.e. 0 mapped to +1 and 1 mapped to –1). The mapped bits are added to noise samples from the AWGN core, the even AWGN samples added to the I symbols and the odd AWGN samples added to Q symbols. The resulting n-bit samples are then quantized to 3, 4, and 5-bit soft data and input into the Viterbi802 decoder for BER curves as shown in Figure 12. Traceback depth of 48 is used for rate ½; and traceback depth of 96 is used for rate ¾ and rate 7/8. Best state logic is used in these BER plots. Figure 12 shows results for continuous data and the number of input data points vary from 2**25 at lowest SNR to 2**39 at highest SNR.

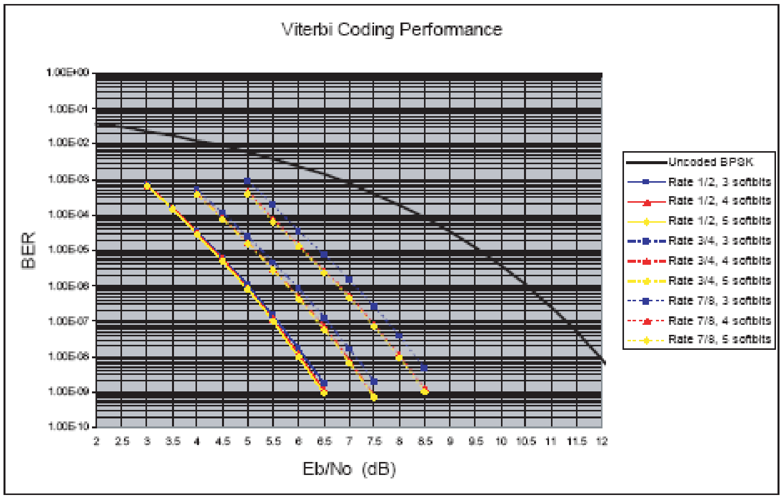


Figure 12: **Viterbi802 BER with Different Soft Bits and Code Rates**

Ordering Information

This Viterbi Decoder core is sold as source code and is provided under the **LogiCORE Core Site License Agreement**. The fixed netlist version of the core is provided under the **SignOnce IP Site License**. A free evaluation version is available from Xilinx DSP marketing or through your Xilinx sales representative.

For part number and pricing information, see the core product page on the Xilinx **IP Center**. To purchase this core, contact your local **sales representative**.

Information on additional Xilinx LogiCORE modules is available on the Xilinx **IP Center**.

Revision History

This table shows the revision history of this document.

Date	Version	Revision
07/09/02	1.0	Initial Xilinx release.
12/10/02	2.0	Added subsection t*bs0 Input and tbd1out, tbquout Outputs*, and 2 related paragraphs. Revised feature bullet entries and the *Add Compare Select* section. Added signals to Figure 1 and to the core pinout tables. Revised *Latency* section. Added sections: *IEEE802.11a Recommendations* and *Hardware Verification*. Added Figure 12.
01/03/03	3.0	Updated copyright date to 2003.
04/23/04	4.0	Updated copyright date to 2004; added Spartan-3 and Virtex-4 to supported FPGA devices.
11/10/04	4.1	Updated document to include clock frequency data for Virtex-II Pro and Virtex-4 under *Performance Characteristics*.