



## CSELT S.p.A

Via G. Reiss Romoli, 274  
I-10148 Torino, Italy  
Phone: +39 011 228 7165  
Fax: +39 011 228 7003  
E-mail: [viplibrary@cse.lt](mailto:viplibrary@cse.lt)  
URL: [www.cse.lt](http://www.cse.lt)

## Features

- Supports Spartan, Spartan™-II, Virtex™, and Virtex™-E devices
- Decoder of convolutional codes
- Customizes VHDL source code available, allowing generation of different netlist versions
- Customized testbench for pre- and post-synthesis verification supplied with the module
- Core customization:
  - Convolutional code definition parameters: Code rate; Code generation vectors; Code constraint length
  - Number of input bits per symbol bit (specifies number of quantization levels for soft decoding)
  - Traceback decision depth
  - Radix-2 / radix-4 architecture selection
  - ACS processors sharing factor
  - Optional inclusion of depuncturing unit interface
  - Optional inclusion of stream alignment/BER estimation unit
  - Estimated BER precision

## Applications

- Decoding of convolutional codes

| AllianceCORE™ Facts                                |  |             |
|--|--|-------------|
| Core Specifics <sup>1</sup>                        |  |             |
| Supported Family                                   | Spartan  | Virtex      |
| Device Tested                                      | S40-3  | V50-6       |
| CLBs <sup>2</sup>                                  | 632  | 495         |
| Clock IOBs   | 1  | 1           |
| IOBs <sup>3</sup>                                  | 34   | 34          |
| Performance (MHz)                                  | 23   | 56          |
| Xilinx Tools                                       | M1.5i/M2.1i  | M1.5i/M2.1i |
| Special Features                                   | SelectRAM  | 4 BlockRAMs |
| Provided with Core                                 |  |             |
| Documentation                                      | User Manual  |             |
| Design File Formats                                | EDIF netlist, XNF netlist, VHDL source available extra |             |
| Constraints File                                   | TOP_VITERBI_DEC_n1.ncf                                 |             |
| Verification                                       | VHDL testbench   |             |
| Instantiation Templates                            | VHDL, Verilog  |             |
| Reference Designs & Application Notes              | None   |             |
| Additional Items                                   | None   |             |
| Simulation Tool Used                               |  |             |
| Synopsys VSS                                       |  |             |
| Support  |  |             |
| Design and customization support provided by CSELT |  |             |

### Notes:

1. Data refer to the following customization:
  - Code parameters: rate = 1/2, constraint length = 5, generation vectors = (23)<sub>8</sub>,(33)<sub>8</sub>;
  - 3 input bits per symbol bit;
  - Traceback decision depth = 32;
  - Radix-2 architecture with sharing factor = 4;
  - Depuncturing unit interface present;
  - BER estimation unit present, BER estimate computed on 16 bits.
2. Utilization numbers for Virtex are in CLB slices
3. Assuming all core I/Os are routed off-chip

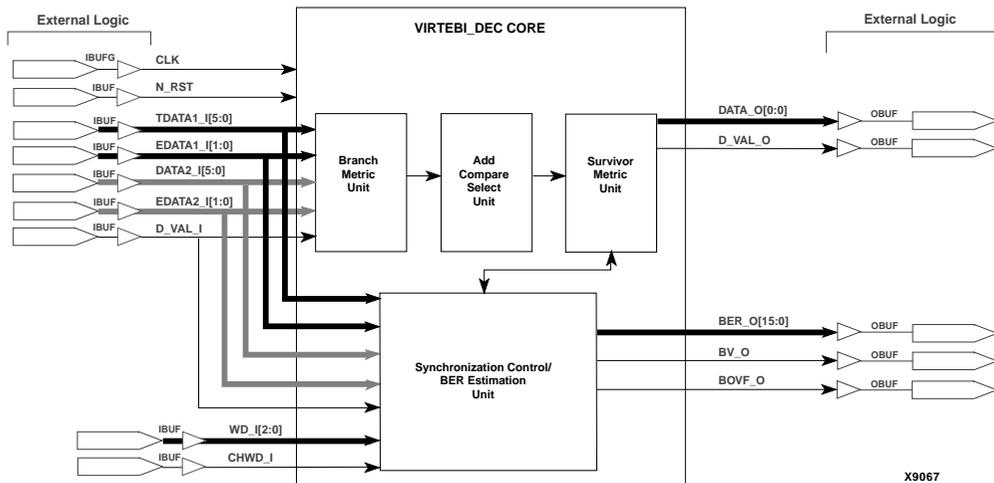


Figure 1: Viterbi Decoder Block Diagram

## General Description

The VITERBI\_DEC core implements a decoder of convolutional codes based on the Viterbi algorithm. It performs decoding as a search of the minimum cost path in a weighted oriented graph, called trellis.

The core decodes a specific convolutional code, with user defined constraint length, rate and generation vectors. User customisable core features also include the number of decoder input bits per symbol bit (used for specifying the number of input quantization levels in case of "soft" decoding) and the decision depth of the decoder traceback algorithm.

The core architecture can be set so as to meet the target design throughput and area requirements. Two basic architectures are available – radix-2 and radix-4. The radix-2 architecture processes one input symbol at a time, while the radix-4 architecture processes two consecutive input symbols at a time. The area-throughput trade-off for both architectures can be modified by defining the ACS processor sharing factor. Dashed-ports in the block diagram are connected only in radix-4 architecture.

The Absolute Maximum ratings, Operating Conditions, DC Electrical Specifications and Capacitances depend on the Xilinx device selected for implementation and can be retrieved from the corresponding Xilinx datasheet.

## Functional Description

The internal architecture of the VITERBI\_DEC core is shown in Figure 1. The decoder is composed of three functional blocks - Branch Metrics Unit (BMU), Add-Compare-Select (ACS), and Survivor Metrics Unit (SMU) - that are

always present in the synthesized instance, and an optional block for BER estimation and BER based stream delineation control. A brief description of the operation of each module follows.

### Branch Metric Unit (BMU)

For each incoming symbol, the radix-2 BMU computes the set of the corresponding branch metrics, i.e. the set of the distances between the incoming symbol and each symbol in the code space. The radix-4 BMU performs the same operation on each pair of consecutive input symbols.

The distances computed by the BMU are linear (Hamming distances). This choice is standard for hard decision decoders in which the input symbol bits are hard 0s and 1s. For soft decision decoders in which the received symbol bits are quantized, nonlinear distances (e.g. Euclidean distances) are normally used; however, non-linearity can be easily moved into the quantizer. That allows using linear distances also in this case, avoiding the implementation of complex computations in the BMU.

The BMU includes an optional depuncturing interface to be used for decoding punctured codes. A code is said to be punctured when symbols in predefined positions of the encoded sequence are erased from the stream before transmission. This way, the overall code rate is lowered, at the cost of reduced error coverage. At the receiver side, an external depuncturing unit signals to the decoder the position of erased symbols, which have equal distances from every possible code symbol.

## Add Compare Select (ACS) unit

The ACS unit is the processing core of the decoder; it computes Viterbi algorithm path metrics through iterative Add-Compare-Select operations. For each decoder state, the ACS unit adds the current path metric of each predecessor state to the corresponding branch metric for the current input symbol, compares the results and selects the smallest among them as the updated path metric of the state. The path selection result, the decision bit, is forwarded to the Survivor Metrics Unit.

The ACS unit is composed of ACS processors, each of which elaborates the path metrics of one state per clock cycle. ACS processor can be either dedicated or shared among states; the former solution maximizes the decoder throughput, while the latter reduces the decoder area. However, ACS processor sharing also reduces the throughput: when each processor serves two (four) states the decoder throughput is also reduced to one half (one quarter) of the clock rate. The architecture and size of the ACS processor also depend on the implemented decoder architecture, i.e., a radix-4 processor must select one path out of 4 and is definitely larger than a radix-2 processor.

## Survivor Metric Unit (SMU)

The SMU elaborates the decision bits from the ACS unit and produces as output the decoded sequence.

The decoded sequence is produced through a traceback on the stored decision bits. Decoding is performed in two phases. In the first phase, paths are traced starting from the minimum cost current state and stepping backwards in time. Decisions about predecessor states for each time step are made in accordance with the stored decision bits for that step. The number of backward steps (*decision depth* L) must be sufficient for allowing the traced paths to converge to one state.

In the second phase, actual decoding takes place. Traceback continues for L further steps from the state found at the end of the first phase. The decoded sequence bits are stored in a LIFO memory. At the end of the second phase the decoded sequence becomes available for output. The decision bits used in the decoding phase are disposed and replaced by those used in the traceback phase.

## Synchronization Control/BER Estimation Unit

This block compares the encoded input data flow with the re-encoded output data flow, giving as a result an estimate of the number of errors inserted by the transmission channel.

The WD\_I input is used to set the dimension of the time window in which errors will be counted. The resulting number of errors will be shown on the BER\_O output bus.

Table 1: Core Signal Pinout

| Signal        | Signal Direction | Description   |
|---------------|------------------|---|
| CLK           | Input            | Master clock  |
| N_RST         | Input            | Asynchronous reset  |
| DATA1_I[5:0]  | Input            | Parallel data input; port size equal to N x Q (N,Q: generics)   |
| DATA2_I[5:0]  | Input            | Parallel data input; port size equal to N x Q (N,Q: generics); port connected only if a radix-4 architecture is selected for implementation |
| EDATA1_I[1:0] | Input            | Erased symbol control; port size is equal to N  |
| EDATA2_I[1:0] | Input            | Erased symbol control; port size is equal to N; port connected only if a radix-4 architecture is selected for implementation                |
| D_VAL_I       | Input            | Data valid input  |
| DATA_O[0:0]   | Output           | Decoded data output; port size is 1 bit for radix-2 architectures, 2 bits for radix-4 architectures   |
| D_VAL_O       | Output           | Data valid output   |
| WD_I[2:0]     | Input            | Window dimension input; port size equal to WDSIZE   |
| CHWD_I        | Input            | Change window dimension input   |
| BER_O[15:0]   | Output           | Estimated BER output; port size is equal to BSIZE   |
| BV_O          | Output           | BER valid output  |
| BOVF_O        | Output           | BER overflow flag   |

**Table 2: Core Parameters (VHDL Generics)**

| Parameter               | Description                                    |
|-------------------------|--|
| N                       | Inverse code rate                              |
| CONST_LEN<br>GTH        | Code constraint length                         |
| POL_GENx<br>(x=0,1,2,3) | Code generation vectors                        |
| Q                       | Number of input quantization bits              |
| L                       | Traceback decision depth                       |
| PAR_IN                  | Radix-2/radix-4 architecture selector          |
| ITERST                  | ACS processor sharing factor                   |
| ESCTRL                  | Depuncturing unit interface instantiation flag |
| BSCTRL                  | BER/Sync control instantiation flag            |
| BSIZE                   | Estimated BER precision                        |
| WDSIZE                  | BER estimation time                            |
| FWSIZE                  | BER control memory word size                   |

## Pinout

Pinout of the core has not been fixed to a specific FPGA I/O allowing flexibility with a user's application. Signal names are shown in the block diagram in Figure 1 and described in Table 1.

## Core Modifications

CSELT provides netlist customized to user's requirements. The VITERBI\_DEC core source code is parametric. Parameters shown in Table 2 are implemented as a set of generics in the synthesizable VHDL source code of the core. Parameters allow the user to specify some architectural and functional features, so as to adapt the netlist to a specific design or application.

## Verification Methods

Extensive functional (pre-synthesis) and timing (post-synthesis) simulation has been performed for different values of the core parameters, using the Synopsys VSS simulator. Simulation scenarios (including data and command files) and parametric test bench used for design verification are provided with the core.

The parametric test bench is composed of a convolutional encoder, a noisy transmission channel and a depuncturing emulator. The input data flow, the noise level of the channel and the puncturing parameters are easily customisable editing some text files.

## Recommended Design Experience

Experience with the Xilinx design flow and convolutional encoding and the Viterbi algorithm is recommended to the users of the netlist version of the core. For the source code version, users should also be familiar with the Synopsys

FPGA synthesis tools (VHDL Compiler, FPGA Compiler) and simulator (VSS).

## Ordering Information

The VITERBI\_DEC core is provided under license by CSELT S.p.A. for use in Xilinx programmable logic devices. Please contact CSELT S.p.A. for information about pricing, terms and conditions of sale.

CSELT S.p.A. reserves the right to change any specification detailed in this document at any time without notice, and assumes no responsibility for any error in this document.

All trademarks, registered trademarks, or servicemarks are property of their respective owners.

## Related Information

### Xilinx Programmable Logic

For information on Xilinx programmable logic or development system software, contact your local Xilinx sales office, or:

Xilinx, Inc.  
2100 Logic Drive  
San Jose, CA 95124  
Phone: +1 408-559-7778  
Fax: +1 408-559-7114  
URL: [www.xilinx.com](http://www.xilinx.com)

For general Xilinx literature, contact:

Phone: +1 800-231-3386 (inside the US)  
+1 408-879-5017 (outside the US)  
E-mail: [literature@xilinx.com](mailto:literature@xilinx.com)

For AllianceCORE™ specific information, contact:

Phone: +1 408-879-5381  
E-mail: [alliancecore@xilinx.com](mailto:alliancecore@xilinx.com)  
URL: [www.xilinx.com/products/logiccore/alliance/tblpart.htm](http://www.xilinx.com/products/logiccore/alliance/tblpart.htm)



## Viterbi Decoder Implementation Request Form

To: CSELT S.p.A.  
 FAX: +39 011 228 7003  
 E-mail: [viplibrary@cse.lt](mailto:viplibrary@cse.lt)

CSELT configures and ships Xilinx netlist versions of the Viterbi Decoder core customized to your specification. Please fill out and fax this form so that CSELT can respond with an appropriate quotation that includes performance and density metrics for the target Xilinx FPGA.

From: \_\_\_\_\_  
 Company: \_\_\_\_\_  
 Address: \_\_\_\_\_  
 City, State, Zip: \_\_\_\_\_  
 Country: \_\_\_\_\_  
 Phone: \_\_\_\_\_  
 FAX: \_\_\_\_\_  
 E-mail: \_\_\_\_\_

### Implementation Issues

1. Coding rate (R): \_\_\_\_\_
2. Constraint Length: \_\_\_\_\_
3. Number of soft input bits: \_\_\_\_\_
4. Length of trace-back: \_\_\_\_\_
5. Data rate (2, 1,  $\frac{1}{2}$ ,  $\frac{1}{4}$ ): \_\_\_\_\_
6. Coding polynomial: \_\_\_\_\_
7. Bit error rate (BER) monitor required? \_\_\_\_\_
8. Required BER estimate precision? \_\_\_\_\_
9. BER estimation window size? \_\_\_\_\_
10. Depuncturing unit interface required? \_\_\_\_\_

### Business Issues

1. Indicate timescales of requirement:  
 \_\_\_\_\_ date for decision  
 \_\_\_\_\_ date for placing order  
 \_\_\_\_\_ date of delivery
2. Indicate your area of responsibility:  
 \_\_\_\_\_ decision maker  
 \_\_\_\_\_ budget holder  
 \_\_\_\_\_ recommender
3. Has a budget been allocated for the purchase?  
 Yes \_\_\_\_\_ No \_\_\_\_\_
4. What volume do you expect to ship of the product that will use this core? \_\_\_\_\_
5. What major factors will influence your decision?  
 \_\_\_\_\_ cost  
 \_\_\_\_\_ customization  
 \_\_\_\_\_ testing  
 \_\_\_\_\_ implementation size
6. Are you considering any other solutions?