

Viterbi Decoders

Introduction

The Hammercores by Altera high performance, soft decision Viterbi decoder cores are optimized for Altera® FLEX® 6000, FLEX 10K and APEX™ 20K devices. They are user parameterized to implement any number of standard decoders, or to quickly realize a custom application. The cores are capable of throughput (decoded bits out) of over 50 to 100 Mbps, even for relatively large constraint lengths, such as 7. The cores are able to support many different puncturing rates, based on a mother code of 1/2.

The decoders can also support hard decision decoding, when the number of soft bits is set to two.

Two decoder core types are in the library:

1. VITTOPA – requires external depuncturing, or uses unpunctured codes.
2. VITTOPB – receives data in a symbol by symbol format, internal depuncturing, lower throughput.

The cores are completely implemented in logic, including the traceback circuitry, with no memory required.

There are two supersets of the cores, with a BER rate estimator included.

Several utilities are included, to generate test cases, and analyze the results. This allows you to verify BER performance of the decoders to various channels.

Ports and Parameters

Table 1. Parameters	
Signal Name	Description
N	This is the number of coded bits. For every bit to be encoded, n bits are output. This parameter can be 2, 3, or 4.
L	This is the constraint length. The constraint length can vary between 3 and 9.
SOFTBITS	The number of soft decision bits per symbol. The range is from 2 bits and up. When SOFTBITS is set to 2 bits, the decoder acts as a hard decision decoder, and still allows for erased symbols to be entered as binary "00".
BMGWIDE	This is the precision of the branch metric accumulation. It can be any value, SOFTBITS + 4 bits or greater. The decoder may flag an error when compiling for certain combinations of N and L , and specify a larger number for BMGWIDE. When the decoder is reset, all metrics are set to zero. Once any metric reaches "100...", all metrics are normalized by dividing their values in half.
V	This parameter is the traceback depth. It is typically set to 5L for unpunctured codes, and up to 15L for highly punctured codes. It must be set to 2 or greater.
GA, GB, GC, GD	A total of N generator polynomials will be required, in decimal form. If N is less than 4, the unused polynomials should be set to zero.
RATE	There are six puncturing schemes directly supported by the VITTOPB and VITBERB decoders. The VITTOPB and VITBERB decoders must only be used with punctured codes. Puncturing matrixes are described in <i>Puncturing Schemes</i> . The VITTOPA and VITBERA decoders can support any puncturing scheme allowed by the combination of N and L , as the depuncturing is done external to the decoder. The supported schemes are "0/0" (VITTOPA and VITBERA only), "2/3", "3/4", "4/5", "5/6", and "6/7". The RATE parameter is not applicable to VITTOPA. It is applicable to VITBERA, because the BER estimation unit must take into account (and ignore) de-punctured symbols.

A-WP-HCORES_VITERBI

Date: February 2000

1

Ports - VITTOPA

<i>Table 2. Input Signals</i>	
Signal Name	Description
SYSCLK	SYSCLK is the main system clock. N symbols are read in at each rising edge of the clock, and one decoded bit is output, once the traceback buffer has been filled.
RESET	The entire decoder is asynchronously reset when the RESET signal is asserted high. All branch metrics are reset to zero, and the traceback buffer is cleared.
ENABLE	The operation of the decoder is enabled when this signal is asserted high. When it is low, no symbols are latched into the core, and no branch metrics are updated, or any tracebacks calculated.
RR[(n*softbits)..1]	This bus takes in one n symbols, each softbits wide per clock. The first symbol received will occupy the most significant bits of rr[], and the last (nth) symbol received will occupy the least significant bits of rr[]. This can be seen in the test cases generated by the VVEC.EXE utility. Erased (de-punctured) symbols are equal to zero. An encoded '1' is negative (i.e. "1XX..") and an encoded '0' is positive (i.e. "0XX..").

<i>Table 3. Output Signals</i>	
Signal Name	Description
SYMBOL	This signal contains the value of the current decoded bit. Decoded bits will be read out after a number of clocks: the depth of the traceback, plus the pipeline depth of the decoder.
NORMALIZE	This signal will be asserted high for 1 clock cycle, whenever the branch metrics are normalized.

Ports - VITTOPB

<i>Table 4. Input Signals</i>	
Signal Name	Description
SYSCLK	SYSCLK is the main system clock. One symbol is read in at each rising edge of the clock, and one decoded bit is output, once the traceback buffer has been filled.
RESET	The entire decoder is asynchronously reset when the RESET signal is asserted high. All branch metrics are reset to zero, and the traceback buffer is cleared.
ENABLE	The operation of the decoder is enabled when this signal is asserted high. When it is low, no symbols are latched into the core, and no branch metrics are updated, or any tracebacks calculated.
RR[softbits..1]	This bus takes in one symbol per clock. An encoded '1' is negative (i.e. "1XX..") and an encoded '0' is positive (i.e. "0XX..").

Output Signals

Table 5. Output Signals

Signal Name	Description
SYMBOL	This signal contains the value of the current decoded bit. Decoded bits will be read out after a number of clocks: the depth of the traceback, plus the pipeline depth of the decoder.
NORMALIZE	This signal will be asserted high for 1 clock cycle, whenever the branch metrics are normalized.
VALID	When high, the SYMBOL output contains a decoded bit. When low, the decoder is disabled while processing depuncturing, and the value of SYMBOL should be ignored.

Ports – VITBERA, VITBERB

VITBERA and VITBERB have the same ports as VITTOPA and VITTOPB, respectively, except that a BER estimation section has been added. Table 6 describes only those ports that deal with the BER section.

Table 6. Input Signals

Signal Name	Description
PERIOD[24..1]	This bus contains the number of decoded bits that the BER measurement is based on. The 24 bit width allows BER measurements with a granularity as small as approximately 1.5×10^8 . The value may be changed at any time, without affecting the current BER measurement, and the new value of PERIOD[] will affect the next BER measurement.

Table 7. Output Signals

Signal Name	Description
BERERR[16..1]	This bus contains the number of bit errors estimated in the last BER measurement period. It is latched at the end of each measurement period.
NUMERR[16..1]	This bus contains the number of bit errors estimated the current measurement period. It is updated with each error detected.

Code Selection

Recommended Generators

The following tables list maximum distance codes when parameter n is 2, 3, and 4, with constraint lengths from $L = 4$ to $L = 9$. The generators are given in both octal and decimal values. The core parameters are entered in decimal form.

Table 8. Maximal Distance Codes for $n = 2$

Constraint Length L	Generators in Octal		Generators in Decimal	
	GA	GB	GA	GB
4	15	17	13	15
5	23	35	19	29
6	53	75	43	61
7	133	171	91	121
8	247	371	167	249
9	561	753	369	491

Table 9. Maximal Distance Codes for $n = 3$

Constraint Length L	Generators in Octal			Generators in Decimal		
	GA	GB	GC	GA	GB	GC
4	13	15	17	11	13	15
5	25	33	37	21	27	31
6	47	53	75	39	43	61
7	133	145	175	91	101	125
8	225	331	367	149	217	247
9	557	663	711	367	435	457

Table 10. Maximal Distance Codes for $n = 4$

Constraint Length L	Generators in Octal				Generators in Decimal			
	GA	GB	GC	GD	GA	GB	GC	GD
4	13	15	15	17	11	13	13	15
5	25	27	33	37	21	23	27	31
6	53	67	71	75	43	55	57	61
7	135	135	147	163	93	93	103	115
8	235	275	313	357	157	189	203	239
9	463	535	733	745	307	349	475	485

Puncturing Scheme

All punctured codes shown below are based on a mother code of rate $\frac{1}{2}$. The VITTOPA and VITBERA decoders require the user to de-puncture the received data stream external to the decoder, and input the data into the decoder n symbols at a time. The VITTOPB and VITBERB decoders input the received symbols one symbol at a time, and de-puncture the data internally. The puncturing rates supported are 0 (unpunctured), $\frac{2}{3}$, $\frac{3}{4}$, $\frac{4}{5}$, $\frac{5}{6}$, and $\frac{6}{7}$. The table below shows the puncturing scheme used, by rate.

Table 11. Puncturing Scheme Used by Viterbi Cores							
Rate	Puncturing Scheme						
	Bit	Multiplier					
2/3	CA	1	0				
	CB	1	1				
3/4	CA	1	0	1			
	CB	1	1	0			
4/5	CA	1	0	0	0		
	CB	1	1	1	1		
5/6	CA	1	0	1	0	1	
	CB	1	1	0	1	0	
6/7	CA	1	0	0	1	0	1
	CB	1	1	1	0	1	0

CA refers to the most significant (first transmitted bit, first received symbol), and CB refers to the least significant (last transmitted bit, last received symbol).

Other Punctured Codes

Other punctured codes, either with different puncturing matrixes for codes based on a mother rate of $\frac{1}{2}$, or punctured codes based on mother rates of $\frac{1}{3}$ or $\frac{1}{4}$, can be implemented by the user. The VITTOPA and VITBERA decoders take in de-punctured symbols, so the user can de-puncture the received stream externally to the decoders. If the state machine controlling the de-puncturing requires to halt the decoder while zeros are inserted into the symbol stream, the ENABLE port on the decoder can be de-asserted.

VITTOPA, VITBERA Core

The VITTOPA decoder takes in n symbols per clock, and returns one decoded bit per clock. The system clock rate of the core is the same as the output bit rate.

If a punctured code is being used, the user must de-puncture the received bits externally to the core. This will require two clocks in the system; a faster received bits clock, and a slower decoder clock. Even if the code is unpunctured, the user must still combine n symbols into a parallel symbol vector (see Ports and Parameters section), and present the symbol vector to the decoder with every decoder clock.

When testing the VITTOPA decoder core with a punctured code, erased symbols are entered as zero.

Setting Compilation Options

The decoders should be compiled with **Global Logic Synthesis** set to FAST. Alternately, **Global Logic Synthesis** set to NORMAL, can be used, which will ease the fitting into a device, at the expense of size and performance.

Example Decoder Design - VITTOPA

The following steps are used to design a rate 1/2, constraint length 5 Viterbi decoder.

The parameters are:

```
N = 2
SOFTBITS = 3,
L = 5
GA = 19
GB = 29
GC = 0
GD = 0
V = 35
BMGWIDE = 10
RATE = "0/0"
```

The MAX+PLUS II™ software was used to compile the top level file, **VITTOPA.TDF**. The targeted device was a 10K30ETC144-1.

The design required 1475 LCs, and had a throughput of 64.5 Mbps.

Example Decoder Design - VITTOPA

The following steps are used to design a rate 1/2, constraint length 7 Viterbi decoder.

The parameters are:

```
N = 2
SOFTBITS = 3,
L = 7
GA = 91
GB = 121
GC = 0
GD = 0
V = 35
```

BMGWIDE = 12

RATE = "0/0"

The MAX+PLUS II software was used to compile the top level file, **VITTOPA.TDF**. The targeted device was a Altera 10K200SQC240-1.

The design required 6293 LCs, and had a throughput of 44.1 Mbps.

Example Decoder Design - VITBERA

The rate $\frac{1}{2}$, constraint length 5 Viterbi decoder from the first example was compiled with the BER estimation section.

The MAX+PLUS II software was used to compile the top level file, **VITBERA.TDF**.

The design required 1660 LCs, and had a throughput of 58.5 Mbps.

VITTOPB, VITBERB Core

The VITTOPB decoder takes in one symbol per clock, and returns one decoded bit per clock. The system clock rate of the core is not the same as the output bit rate – as the decoder de-punctures the incoming data automatically, it has to periodically disable the decoder output, to match the input and output rates. The advantage of this approach over external de-puncturing is that only one clock – the received punctured clock rate – is needed.

Setting Compilation Options

The decoders should be compiled with **Global Logic Synthesis** set to FAST. Alternately, **Global Logic Synthesis** set to NORMAL, can be used, which will ease the fitting into a device, at the expense of size and performance.

Example Decoder Design - VITTOPB

The following steps are used to design a rate $\frac{1}{2}$ (punctured rate $\frac{2}{3}$) constraint length 6 Viterbi decoder.

The parameters are:

N = 2

SOFTBITS = 5,

L = 6

GA = 43

GB = 61

GC = 0

GD = 0

V = 40

BMGWIDE = 12

RATE = 2/3

The MAX+PLUS II software was used to compile the top-level file, **VITTOPA.TDF**. The targeted device was an Altera 10K100EQC208-1.

The design required 3574 LCs, and had a maximum clock rate of 53.5 Mbps.

Example Decoder Design - VITBERB

The rate $\frac{1}{2}$, constraint length 6 Viterbi decoder from above was compiled with the BER estimation section.

The MAX+PLUS II software was used to compile the top level file, **VITBERB.TDF**.

The design required 3783 LCs, and had a throughput of 49.5 Mbps.

Core Testing

Generating Test Vectors

A test vector generation program, **VVECE.EXE**, is used to create test cases for both the VITTOPA and VITTOPB decoders. The VITBERA and VITBERB decoders can also be tested with the test cases generated, by adding the busses for the BER monitor in the simulator.

The VVECE utility uses the following inputs:

- Bits : The number of bits to be coded for the testcase
- Eb/No : The SNR for the AWGN channel used in the testcase.
- Softbits : The parameter softbits of the decoder.
- N : The parameter n of the decoder
- L : The parameter L of the decoder
- V : The parameter V of the decoder
- Rate : The parameter rate of the decoder. This input is given as a value, rather than a string. Valid inputs are: 0, 23, 34, 45, 56, and 67.
- Generator Polynomials, in decimal.

Running the program without a complete set of inputs will display the required inputs. When VVECE is run with a complete parameter list, it will output the following results:

1. List of generator polynomials, in decimal.
2. Actual BER for the test case. As the error locations are randomly generated, the actual BER may differ slightly from the expected BER for a given SNR. The number of bit errors will also be displayed.

The BER rate may differ from what the user is expecting, as there are many different modulation formats, that have differing uncoded BER rates. The BER rates are output so that the user can verify the decoder performance to BER in.

The program will generate the following files:

Note: For VITTOPA, n symbols are input per clock. These are known as symbol vectors.

1. **TRANSBIT.TXT** – these are the randomly generated bits coded in the test case.
2. **A_TXSYM.TXT** – these are the encoded symbol vectors for the test case, for the VITTOPA decoder. The transmitted symbols are ‘1’ and ‘0’.
3. **A_RXSYM.TXT** – these are the encoded symbol vectors, with the AWGN channel added, for the VITTOPA decoder. Each symbol is of `softbits` precision.
4. **A_ERRLOC.TXT** – these are the locations by symbol vector, of the received errors. More than one error can be indicated per symbol vector, as n symbols are received per clock.
5. **B_TXSYM.TXT** – these are the encoded symbols for the test case, for the VITTOPB decoder.
6. **B_RXSYM.TXT** – these are the encoded symbols, with the AWGN channel added, for the VITTOPB decoder.
7. **B_ERRLOC.TXT** – these are the locations by symbol, of the received errors

Running the Test Case

To run a test case, the appropriate vector (*.VEC) file for the decoder to be tested must first be loaded into the Altera simulator.

To simulate in the MAX+PLUS II software, open a simulator window, and use **Inputs/Outputs** (File Menu) to point to the **VITTOPA.VEC** or **VITTOPB.VEC** file that was created in the directory where **VVECE.EXE** is in. The simulator will convert the *.VEC file into a *.SCF file.

Simulate. When the simulation is complete, open the *.SCF file, and use **Create Table File** (File Menu) to generate a text output of the simulation. This text output will be read by either **VTBLAA.EXE** or **VTBLBB.EXE** (explained in the next section).

Testing VITBERA and VITBERB

The VITBERA and VITBERB decoders can be tested using the **VITTOPA.VEC** and **VITTOPB.VEC** files, respectively. The files are imported into the simulator by the method explained above. The additional three busses, **PERIOD[24..1]**, **NUMERR[16..1]**, and **BERERR[16..1]**, are then added using **Enter Nodes from SNF** (Nodes Menu).

To analyze the results using **VTBLAA.EXE** and **VTBLBB.EXE**, respectively, the three busses must be deleted from the .SCF file, before creating the table file.

Analyzing Test Results

Each decoder uses a different program to analyze test results. The VITTOPA decoder uses **VTBLAA.EXE**, and the VITTOPB decoder uses **VTBLBB.EXE**.

VTBLAA

The VTBLAA program requires the following parameters:

- Bits – the number of bits to compare

- Traceback – the `v` parameter of the decoder
- Number of Coded Bits – the `n` parameter of the decoder.

Running VTBLAA without a complete parameter list will display the required inputs. When a valid input list is given to the program, it will display the following outputs:

```
The first symbol is at X ns
Reading Simulation File
Writing RESULTS File
Number of Errors Found is Y
Output BER is Z
```

Where X, Y and Z are the results calculated during analysis of the test case output.

The **ERRLOCA.TXT** file is output, which contains the location of errors in the decoded output compared to the original symbols in the **TRANSBIT.TXT** file.

VTBLBB

The VTBLBB program requires the following parameters:

- Bits – the number of bits to compare
- Traceback – the `v` parameter of the decoder
- Rate – the `rate` parameter of the decoder

Running VTBLBB without a complete parameter list will display the required inputs. When a valid input list is given to the program, it will display the following outputs:

```
The first symbol is at X ns
Reading Simulation File
Writing RESULTS File
Number of Errors Found is Y
Output BER is Z
```

Where X, Y and Z are the results calculated during analysis of the test case output.

The **ERRLOCB.TXT** file is output, which contains the location of errors in the decoded output, compared to the original symbols in the **TRANSBIT.TXT** file.

BER Monitor

There two decoders with a BER monitor, VITBERA and VITBERB. The BER monitor uses a re-encode and compare approach for estimating errors.

In cases where the SNR is sufficiently high to allow the decoder to decode an error-free output, the BER estimation will be very close to the actual channel BER. When the decoder is not decoding an error-free output, the estimated BER will be higher than the actual channel BER, as the state of the re-encode section will be wrong for multiple estimations.

Appendix A. Top Level Wrappers

For each of the four decoders, there is an unencrypted top level wrapper, which can be used easily modify the parameters of the core. As the top level wrappers are in source form, they can also be used to create symbol (*.SYM) and include files (*.INC).

The names of the wrappers are:

TOP_LEVEL_VITTOPA.TDF

TOP_LEVEL_VITTOPB.TDF

TOP_LEVEL_VITBERA.TDF

TOP_LEVEL_VITBERB.TDF

As an example, the source code for TOP_LEVEL_VITTOPA is shown below:

```

FUNCTION vittopa (sysclk, reset, enable, rr[(n*softbits)..1])
  RETURNS (normalize, symbol);

PARAMETERS
(
  n = 2,
  softbits = 8,
  L = 5,
  ga = 19,          -- 1 0 0 1 1 (23 octal)
  gb = 29,          -- 1 1 1 0 1 (35 octal)
  gc = 0,
  gd = 0,
  bmgwide = 17,
  v = 47
);

subdesign top_level_vittopa
(
  sysclk, reset, enable : INPUT;

```

```
rr[(n*softbits)..1] : INPUT;
normalize, symbol : OUTPUT;
)

BEGIN

    (normalize, symbol) = vittopa (sysclk, reset, enable,
rr[(n*softbits)..1])

    WITH
(n=n,softbits=softbits,L=L,ga=ga,gb=gb,gc=gc,gd=gd,
                                bmgwide=bmgwide,v=v);

END;
```



101 Innovation Drive
San Jose, CA 95134
(408) 544-7000
<http://www.altera.com>

Copyright © 2000 Altera Corporation. Altera, FLEX, MAX+PLUS II, and AMPP are trademarks and/or service marks of Altera Corporation in the United States and other countries. Other brands or products are trademarks of their respective holders. The specifications contained herein are subject to change without notice. Altera assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Altera Corporation. Altera customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services. All rights reserved.