



OpenCores.Org

**“HDBN”**  
**HDB3/HDB2/B3ZS**  
**Encoder / Decoder**  
**IP core**  
**Specification**

*Author: Allan Herriman*  
*allanh@opencores.org*

**Rev. 0.1**  
**April 29, 2003**

*This page has been intentionally left blank.*

## Revision History

<b>Rev.</b>	<b>Date</b>	<b>Author</b>	<b>Description</b>
0.1	29/4/03	Allan Herriman	First Draft

# Contents

<b>INTRODUCTION.....</b>	<b>1</b>
FEATURES.....	1
NOTES .....	1
ACRONYMS .....	2
<b>ARCHITECTURE.....</b>	<b>3</b>
<b>OPERATION .....</b>	<b>4</b>
<b>REGISTERS.....</b>	<b>5</b>
<b>CLOCKS.....</b>	<b>6</b>
<b>GENERIC.....</b>	<b>7</b>
<b>IO PORTS.....</b>	<b>8</b>
<b>APPENDIX A .....</b>	<b>9</b>
<b>APPENDIX B .....</b>	<b>10</b>
<b>INDEX.....</b>	<b>12</b>

# 1

---

## Introduction

This “core” is actually two cores – an HDB3/HDB2/B3ZS Encoder that converts NRZ data into P and N pulses according to ITU-T G.703, and a HDB3/HDB2/B3ZS Decoder that converts P and N pulses into NRZ data according to ITU-T G.703.

Note: HDB2 and B3ZS are different names for the same encoding.

HDB3 is typically used to encode data at 2.048 (E1), 8.448 (E2) and 34.368Mb/s (E3)

B3ZS is typically used to encode data at 44.736Mb/s (T3)

(These encodings don't have much application apart from those particular data rates.)

### Features

- HDB3 / HDB2 selected by a generic.
- Code Error output on decoder.
- P and N outputs (on encoder) or inputs (on decoder) may be active high or active low, selected by a generic.
- P and N outputs on encoder may be controlled to be “full width” (NRZ) or “half width” (RZ) to suit the external LIU.

### Notes

- The encoder does **not** include pulse shaping circuitry to meet the pulse mask of G.703. An external LIU (Line Interface Unit) will be required. (For non-critical applications this may be as simple as a 74ACT244, a few resistors and a pulse transformer.)
- The decoder does **not** include clock and data recovery. An external Slicer and CDR circuit will be required (typically in the LIU chip).

- Many modern LIU chips have a HDB3 encoder / decoder built in (e.g. the LXT350 from Intel) so this core may not be needed.

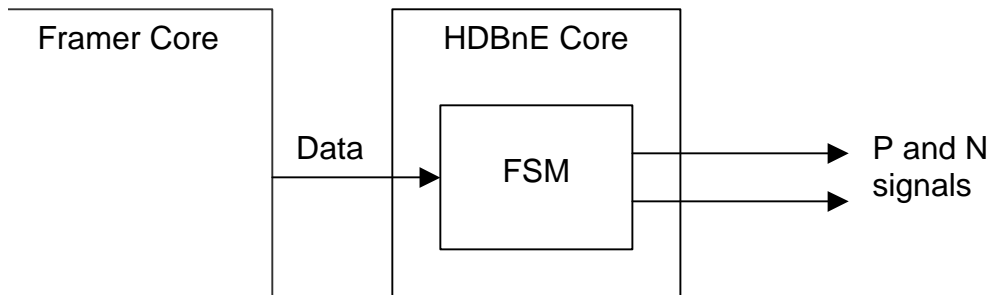
## Acronyms

- HDB3 High Density Bipolar, order 3
- HDB2 High Density Bipolar, order 2
- B3ZS Bipolar with 3 Zero Substitution
- AMI Alternate Mark Inversion
- NRZ Non-Return to Zero
- RZ Return to Zero
- LIU Line Interface Unit

# 2

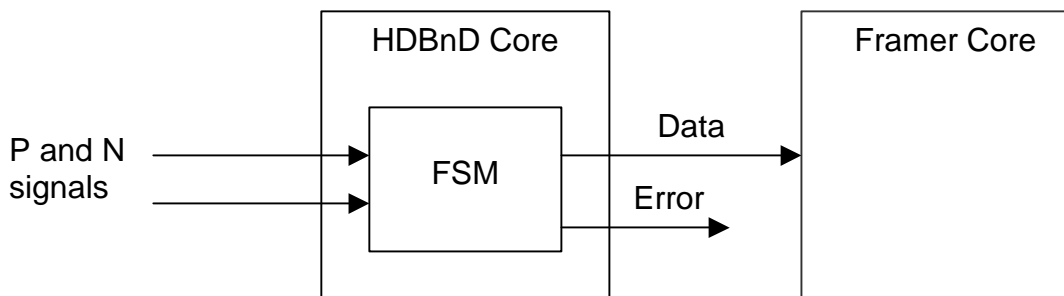
## Architecture

The HDBNE encoder core is just a small state machine with a single clock.



**Figure 1 Encoder Architecture**

The HDBND decoder core is also just a small state machine with a single clock.



**Figure 2 Decoder Architecture**

# 3

---

## Operation

The encoder and decoder cores do not require any initialization or configuration to work.

They don't even require a reset signal (for synthesis), as the internal state machines are inherently free from lockup. A reset signal is provided however, and it should be connected to the system reset.

The reset signal *must* be driven during simulation, in order to clear some simulation-only lockup states associated with uninitialised signals.

The decoder may assert its "code\_error" output when it detects an error condition on its input. There will typically be one or more bit errors on the data output at about the same time.

No action needs to be taken when this occurs (from the point of view of the decoder) however the "application" may wish to use an external programmable counter/timer core to count these errors for performance monitoring purposes.



# 4

---

# Registers

There are no registers. There is no software interface. There is nothing to configure and nothing to monitor.

# 5

## Clocks

Name	Source	Rates (MHz)			Remarks	Description
		Max	Min	Resolution		
clk_i		-	-	-	-	Data clock

Table 1: List of clocks

The encoder and decoder cores each have a single clock. All inputs and outputs of the core are synchronous to this clock. Only the rising edge of the clock is used internally.

The outputs of the cores come straight from the Q outputs of flip flops inside the core.

The frequency of the clock must be the same as the bit rate, assuming the clock enable input to the core is always '1'. Higher speed clocks may be used if the clock enable is controlled appropriately, e.g. E1 rates may be supported with a 4.096MHz clock if the clock enable is '1' every second clock.

HDB3 is typically used to encode data at 2.048 (E1), 8.448 (E2) and 34.368Mb/s (E3)

B3ZS is typically used to encode data at 44.736Mb/s (T3)

Note: the internal logic is very simple, and it should be easy to get the cores to work at >100MHz clock speeds, even in FPGAs with a modest speed grade.

The encoder clock will typically have to meet the rate error and jitter requirements of ITU-T G.703, G.823, etc.

The decoder clock will typically come from a clock and data recovery circuit.

# 6

## Generics

This section specifies the core generics (a.k.a. parameters).

The generics for the encoder and decoder cores are the same.

Generic	Type	Default Value	Description
EncoderType	integer range 2 to 3	3	3: HDB3 2: HDB2/B3ZS
PulseActiveState	std_logic	'1'	Active state of P and N ports '0' => P and N ports are active low '1' => P and N ports are active high

Table 2: List of generics

## 7

# IO Ports

This section specifies the cores' IO ports.

## *HDBNE Encoder*

Port	Type	Direction	Description
Reset_I	Std_logic	In	active high async reset
Clk_I	Std_logic	In	rising edge clock
ClkEnable_i	Std_logic	In	active high clock enable
Data_I	Std_logic	In	active high data input
OutputGate_i	Std_logic	In	'0' forces P and N to not PulseActiveState (synchronously, but ignoring ClkEnable)
P_o	Std_logic	Out	encoded +ve pulse output (see PulseActiveState generic)
N_o	Std_logic	Out	encoded -ve pulse output (see PulseActiveState generic)

Table 3: List of IO ports for Encoder

## *HDBND Decoder*

Port	Type	Direction	Description
Reset_I	Std_logic	In	active high async reset
Clk_I	Std_logic	In	rising edge clock
ClkEnable_i	Std_logic	In	active high clock enable
P_I	Std_logic	In	+ve pulse input (see PulseActiveState generic)
N_I	Std_logic	In	-ve pulse input (see PulseActiveState generic)
Data_o	Std_logic	out	active high decoded data output
CodeError_o	Std_logic	out	active high error indicator

Table 4: List of IO ports for Decoder

# Appendix A

---

## HDB $n$ Coding Rules

*Note: G.703 is a copyright work of the ITU-T. It is only possible to include a summary here.*

HDB3 and HDB2 are basically modified versions of AMI. AMI (Alternate Mark Inversion) is a method of producing a DC balanced line code for data transmission by alternating each successive '1' bit (mark) between a positive voltage and a negative voltage. DC balanced codes are needed when transformer coupling is used.

The bits:

0101100001      could be (AMI) encoded as:

0+0-+0000-      where '+' is a positive pulse and '-' is a negative pulse.

The same signal could also be encoded with the '+' and '-' voltages swapped and still retain the same meaning. This is a handy feature for a code often used on twisted pair wire – the tech doesn't have to distinguish one wire from the other.

AMI transmits '0' bits as zero volts on the line. A long string of '0' bits will cause a receiver to lose bit synchronization due to the lack of transitions.

HDB3 / HDB2 avoid the loss of synchronization by substituting a different pattern for a long string of '0' bits. They use a pattern known as a *violation* to indicate that the substitution has been made. A violation breaks the AMI rule, so a positive pulse can follow a positive pulse, etc. The receiver can detect the violation and substitute the original string of '0' bits.

HDB3/HDB2 replace a string of four/three zeros with either 000V/00V or B00V/B0V, where V is a '1' violating the AMI rule, and B is a '1' not violating the AMI rule.

Back to our example:

0101100001      could be HDB3 encoded as:

0+0-+000+-      where '+' is a positive pulse and '-' is a negative pulse.

The choice of B or 0 is made to eliminate any DC bias. This is achieved by making the V pulses alternate in polarity, and there will be an odd number of B pulses between consecutive V pulses.

# Appendix B

---

## RZ pulses at the encoder output

Some simpler LIUs require the generated P and N pulse signals to have a particular pulse width (typically exactly half a bit period) in order for the output of the LIU to meet the stringent pulse mask requirements of G.703.

Gating the P and N outputs of the decoder with the clock could do this, but using clocks as logic signals is generally frowned upon, and the clock signal typically wouldn't have the required duty cycle accuracy.

The OutputGate input was added to the Encoder to allow the P and N outputs to be forced to their inactive state independently of the ClkEnable input. This can be used (with the ClkEnable input) to produce RZ pulses at the P and N outputs, as shown in Figure 4.

Some care will need to be taken with the phase of the clocks if the data input to the encoder core is generated from a 2.048MHz clock domain.

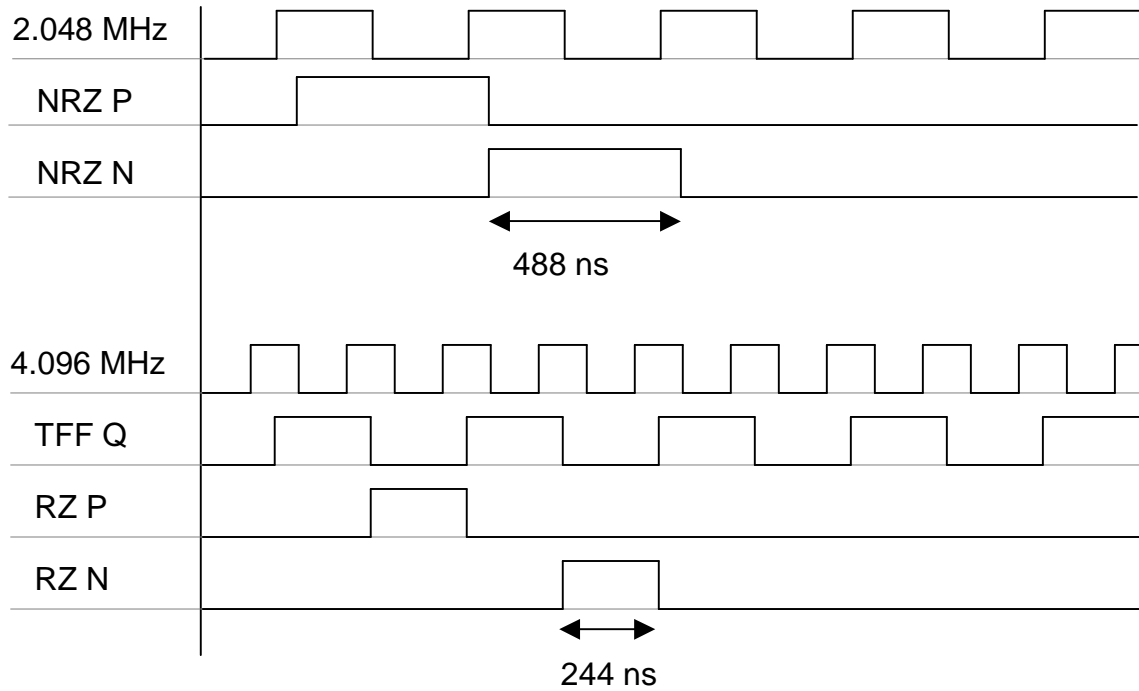


Figure 3 Example of NRZ and RZ waveforms on P and N (not to scale)

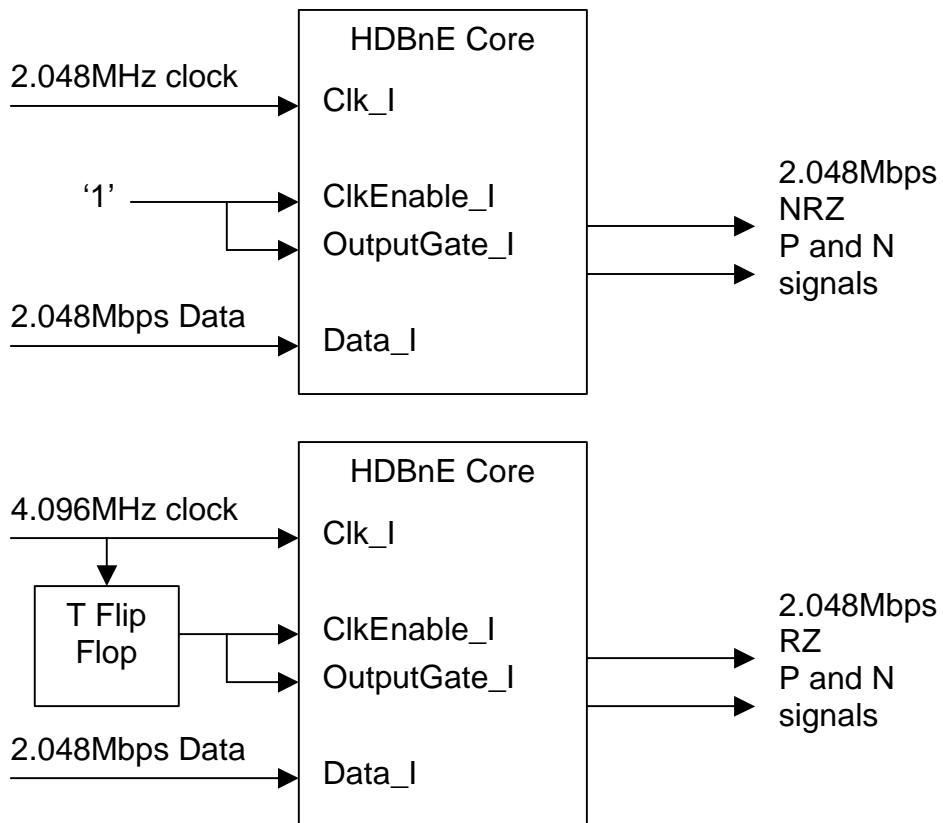


Figure 4 Generating NRZ and RZ pulses at the Encoder output

# Index

---

This section contains an alphabetical list of helpful document entries with their corresponding page numbers.