The OpenCores 8b10b_encdec Project

Version 1.0 – First Release
10.03.2006

8b/10b Coding Background

The 1983 IBM published paper "A DC-Balanced, Partitioned-Block, 8b/10b Transmission Code" by A. X. Widmer and P. A. Franaszek presented the background and methods for generating a 10-bit coded representation of an 8-bit value. This method, developed primarily for use in fiber optic links, was subsequently issued a patent, number 4,486,739, in 1984 which has now expired, so we're free to incorporate it in our designs.

8b/10b has been widely adopted by a variety of high speed data communication standards used today and should prove ever more useful for FPGA-based designs as clock speeds and I/O capabilities increase.

About the OpenCores "8b10b_encdec" Project

Basically, by providing a free open source VHDL implementation of both the 8b-to-10b encoder and 10b-to-8b decoder we hope that many circuit designers can benefit from the ingenuity of Widmer and Franaszek nearly a quarter of a century ago.

Our research has shown that there are a number of commercially available cores today. All of the major FPGA vendors offer licensable 8b/10b cores tailored to their particular family of devices. A number of independent IP developers offer core licenses as well. The goal of our OpenCores project is to provide the foundation for understanding and adopting 8b/10b for a variety of data communications challenges. Our goal was to provide a project that was as close to the original 8b/10b specification as possible. Therefore, we have not currently implemented any optional features such as error checking that might be found in commercial cores. Some users may wish to add additional features to the basic core design. We have therefore adopted the GNU GPL license so that improvements and enhancements can be shared. If you have issues with the GPL license or if you need support, you should consider one of the commercial core vendors.

8b/10b References

This text document is not intended to provide a detailed understanding of the 8b/10b coding methodology, but rather a useful description of the VHDL modules that make up the project. There are a number of sources that describe the coding methods in detail including:

- http://en.wikipedia.org/wiki/8B10B
  - This is a good place to start with a simple description plus links to the original publication and the patent.

- "A DC-Balanced, Partitioned-Block, 8b/10b Transmission Code" by A. X. Widmer and

P. A. Franaszek

  - This is the original IBM publication. A very technical document that provides more that most want to know about 8b/10b coding. However, this is where it all began.

- United States Patent Number 4,486,739, December 4, 1984
  - This is the patent document. Note that our project references the figures contained in the patent, so you probably would like to have this on hand. You can purchase a downloadable PDF very inexpensively from the US Patent Office.

- Cypress CYP15G0401TB Data Sheet, Document #: 38-02112, Cypress Semiconductor
  - This document, describing a commercial Cypress physical layer device that incorporates 8b/10b, does contain the complete 8b/10b coding tables. This is very useful for reference during verification.

- Actel, Altera, Lattice and Xilinx all have on-line documents that describe commercially available 8b/10b cores specifically for their devices. These are useful references for investigating additional features found in commercial cores.

8b10b_encdec Project Description

There are four VHDL modules in the 8b10b_encdec project::

- 8b10b_enc.vhd
- 8b10b_dec.vhd
- enc_8b10b_TB.vhd
- encdec_8b10b_TB.vhd

8b10b_enc.vhd

This is the encoder module. It is synthesizable VHDL'93. It accepts an 8-bit parallel raw (unencoded) data character consisting of bits HI, GI, FI, EI, DI, CI, BI, AI. AI is the least significant bit. There is also an input bit, KI, that indicates that the character input should be encoded as one of the 12 allowable control, or “K”, characters. All of the inputs are synchronized to an input clock, “SBYTECLK” as follows:

> Bits KI, HI, GI and FI are latched internally on the falling edge of the SBYTECLK.
> Bits EI, DI, CI, BI, AI are latched internally on the rising edge of the SBYTECLK

The encoding operation is as described in the original paper and patent. The MS 3 character bits are encoded by an internal 3b/4b encoder and the disparity, or difference in 1's and 0's is determined. The 5 LS bits are encoded by an internal 4b/5b encoder depending upon the disparity value from the previous 3b/4b coding. Likewise, subsequent coding depends upon the “running disparity value” from the previous coding.

The staggered operation of the encoder was designed so that, a serial bit stream could be contiguously output by an associated serializer as the encoding took place, therefore speeding up the process and reducing the need for buffering.

8b10b_dec.vhd

This is the decoder module. It is synthesizable VHDL'93. It accepts a 10-bit parallel encoded character consisting of bits AI, BI, CI, DI, EI, II, FI, GI, HI, JI where AI is the least significant bit. An associated clock, RBYTECLK, latches the 10-bit character on the falling edge. The 8b10b_dec module decodes the 10-bit code into an unencoded 8-bit character along with an associated "KO" bit to indicate if the character output is one of the 12 allowable control, or "K", characters.

enc_8b10b_TB.vhd

This is the testbench file, written in non-synthesizable VHDL for the encoder by itself. It generates the test clock, TBYTECLK, as a 10MHz source for the encoder and a reset, TRESET, signal for initialization.

The testbench output consists of 8 bits of character output, TAO through THO, along with the TKO to indicate a control character. All outputs are valid with the rising edge of TBYTECLK.

First, the 12 "K" characters are generated in sequence from K28.0 to K30.7, then the 256 data characters are sequentially output from K0.0 to D31.7.

The encoder testbench instantiates the 8b10b_enc encoder and port maps its outputs to the encoder's inputs.

The encoder testbench also port maps the encoder 10-bit output to it's associated inputs, TA through TJ. These are also latched onto signals "tlcharout(9 downto 0)" for convenient observation by a waveform viewer.

encdec_8b10b_TB.vhd

This testbench performs the functions of the encoder testbench and also connects the latched output of the encoder to the input of the decoder. It also uses signals "tdec(7 downto 0)" and "tdeck" to monitor the decoder output for viewing.

Summary

This OpenCores project, 8b10b_encdec should provide a useful building block for digital data communication applications such as high speed connections between FPGA devices. While offering basic functionality, it clearly demonstrates a practical method of coding and decoding according to the original intent of Widmer and Franaszek.

Comments and Suggestions

Please feel free to email your comments and suggestions to:

kboyette@opencores.org