Semiconductor Optoelectronics
ISSN: 1001-5868

16B/20B ENCODER/DECODER DESIGN AND IMPLEMENTATION ON A XILINX FPGA

Mr. Chandra Shekar P[1], Dr. Basavaraj L[2], Mrs. Chaya P[3]
[1]Research Scholar and Assistant. Professor, Dept. of ECE
ATME College of Engineering, Mysuru, Karnataka, India
[2]Professor, Dept. of ECE
ATME College of Engineering, Mysuru, Karnataka, India
[3]Assistant. Professor, Dept. of ISE
GSSSIETW, Mysuru, Karnataka, India

Abstract - This paper details the 8b/10b - line code that maps 8-bit words to 10-bit symbols in VHDL implementation of a fiber channel byte-oriented transmission encoder and decoder in a Xilinx Spartan 3AN FPGA. Xilinx FPGAs are the lowest power FPGAs available today and can be utilized in any network design where reliable point-to-point transceivers are required. Spartan 3AN FPGAs utilize the patented Fast Zero Power™ (FZP) design technique to simultaneously deliver high performance and low power consumption. These devices offer pin-to-pin delays of less than 5.0 ns, and less than 100 µA of standby current (approximately 1/3 of the power consumed by other competing FPGAs/CPLDs at f_{MAX}). The work is carried out by using Xilinx SPARTAN 3AN kit consisting of Xilinx Spartan 3AN FPGA, Xilinx Cool runner CPLD and SPI Flash Memory with a simulation tools modelsim and the programming is done by Xilinx ISE with support of chipscope pro used for FPGA debugging and analysis. It allows engineers and developers to monitor and analyze the internal signals and states of an FPGA design in real-time, which is incredibly useful for debugging complex digital systems.
Index Terms **– VHDL, Spartan FPGA, transceivers, low power, FPGAs/CPLDs**

1. Introduction to 16b/20b Encoder/Decoder

Today binary codes are used in transmission schemes across fiber optic links. The binary on/off mode provides a transmission scheme ideal for high-speed local area networks and computer links. The 8b/10b data transmission scheme has become the standard for high-speed serial links today. An 8b/10b module provides byte synchronization and the encode/decode scheme for fiber channel communication links. The 8b/10b scheme is part of the physical network layer and can be utilized in any gigabit Ethernet, ATM, wireless or fiber optic transmission link. Many communication systems today transmit information in the form of packets with a defined field structure for both communication and error control. The 8b/10b encoding scheme translates byte-wide data of random "1s" and "0s" into a 10-bit serial data stream. The encoding mechanism generates a balanced bit stream, in which an

equal number of bit transitions occur between logic High and a logic Low level. The 8b/10b encoding technique provides a DC balanced code that will optimize the coding efficiency, clock recovery, error detection, and suitability for ring or point-to-point topologies. A16b/20b transmission scheme incorporates the idea of the 8b/10b transmission code by combining two 8b/10b modules side-by-side. With 16b/20b encoding, a 16-bit word can be encoded and transmitted serially as shown in Figure 1.

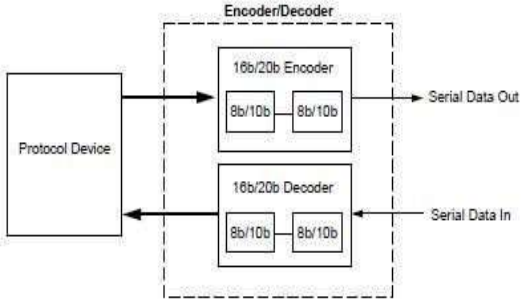

Figure 1: 16b/20b encoding, a 16-bit word

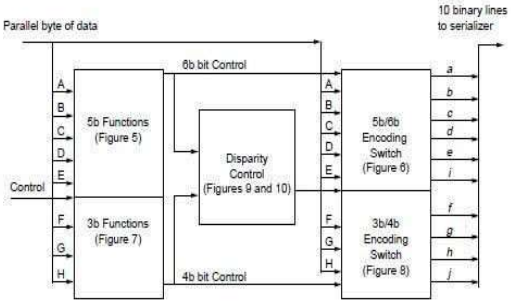

Figure 2: 8b/10b Encoding Block Diagram

The 8b/10b code provides many advantages for fiber optic and electromagnetic wire links. High-gain fiber optic receivers need an AC coupling stage near the front end as well as simplified control of transmitter level, receiver gain, equalization, and maintenance of precise signal power. This becomes more important at high data rates to maintain constant byte rate and reduce redundancy checks to each byte. Lower signaling rates of the 8b/10b code can be utilized to minimize crosstalk and compensate for an increase in signal-to-noise ratio.

2. 8b/10b Background

The 8b/10b transmission code includes serial encoding and decoding rules, special characters, and error detection. The characters defined by this code ensure that short run lengths and sufficient transitions are present in the serial bit stream to make clock recovery possible at the receiver. For this reason, the 8b/10b encoding scheme has the ability to control the characteristics of each code word by creating a limited change of "0s" and "1s". The encoding greatly increases the likelihood of detecting single or multiple errors during the transmission of data.

Each 8b/10b encoding and decoding module is capable of the following main functions - Error detection, Frame delimitation with data transparency, Clock recovery: good signal transition density helps to find the center of each data bit, DC voltage balancing, Fiber optic, wireless or ATM implementation capabilities

3. Structure of 8b/10b Code

The 8b/10b transmission code includes D-characters (used for data transmission) and K characters (used for control and protocol functions). Parity is monitored in each byte of transmission and both D- and K-characters are based on positive or negative parity. The parity of each code word is selected by the encoder to maintain a balanced running parity in the data

stream. For encoding purposes, each incoming byte of data is partitioned into two sub blocks as shown in Figure 2. The five binary lines, ABCDE, are encoded into six binary lines, abcdei, which follow the rules of 5b/6b encoding. Similarly, the three bits, FGH, are encoded into fghj, which follow the 3b/4b encoding rules. The combination of a 5b/6b and 3b/4b encoder blocks limits error propagation to five bits as well as simplifies the encoder and decoder modules. Both the 5b/6b and 3b/4b encoding rules will be discussed in detail.

Every 10-bit encoded data group has one of three possibilities to help limit the number of consecutive "1s" or "0s" in any 2-code words: Five "1s" and five "0s", Four "1s" and six "0s", Six "1s" and four "0s".

Byte synchronization is accomplished through the use of special characters. Select special characters allow for byte stuffing controlling the number of transitions in a data stream.

4. Checking Disparity

DC balancing is achieved through the use of running disparity. This function controls the number of ones and zeros in a single transmission. This helps balance the DC level between the "1" and "0" voltage level. Running disparity is positive (+) if more "1s" than "0s" have been transmitted and negative (–) when more "0s" than "1s" are transmitted. The running disparity remains unchanged from the previous transmission if the code has an equal number of "1s" and "0s". No distinction is made between 6b and 4b blocks, so they are used to compensate each other. Similarly, the disparity in each 8b/10b module is used to compensate the other in a 16b/20b design. For this design, the disparity out of each 16-bit word data transmission is translated as the running disparity when sending a packet of data. The running disparity is then used as the input disparity when encoding the next 16-bit data word to send. Since the disparity is monitored, the transmitted code is free of any DC component.

Error Detection: Error detection in the 8b/10b code is done in two manners:

Error checking using redundancy: Validates transmission of each packet, in which a higher level OSI protocol layer would detect errors based on start and end delimiters.

Cyclic redundancy checks (CRC): Detects errors on individual 6b or 4b sub blocks based on the rules for 8b/10b encoding.

8b/10b Code Definition: The 8b/10b encoding is accomplished by encoding the ABCDE and FGH inputs as two units, the 5b/6b and 3b/4b, respectively. The 10b/8b decoder module is discussed in detail later, and provides the complementary function of the 8b/10b encoder.

5b/6b Definition: Table 1 shows a few examples of the 5b/6b encoding rules. ABCDE represents the lower five bits of the data to send, DATA_IN [0:4] respectively. The transmitted data, abcdei, is shown with the corresponding alternate data that is sent based on the disparity value. The running disparity, D_{in}, can be either positive (+), negative (–), or don't care (x). The disparity out of the encoded word, D_{out}, is shown with the transmitted data. When the alternate data is sent, disparity out (D_{out}) will be complemented. D_{out} can be either positive (+), negative (–), or not affected (0). For each data byte, the bit encoding shows the category each incoming data byte falls into according to the number of zeros and ones. Based on the bit encoding the incoming data bits, ABCDE are translated. If the bit encoding column is left blank, then the data bits do not change and the added i bit is "0". For example, in Table 1, D.0 has bit encoding classification L04, which represents zero "1s" and four "0s" in the ABCD bits of

the data, D.0. The result encoding is then 011000 when the previous running disparity is (+), now resulting in a current negative (–) disparity out. If the input running disparity is (–) then the alternate data, 100111, is sent and the disparity out is (+). The disparity classification shows which functions are used in generating the disparity out for each data byte. These functions are described in more detail in 8b/10b Design, page 10. Table 1 show examples of non-special characters, where the K control character is held at "0".

Table I: 5b/6b Encoding

	Data to Send						Classification			Transmitted Data							
Name	A	B	C	D	E	K	Bit Encoding[1]	Disparity[1,2,3]	Din	a	b	c	d	e	i	Dout	Alternate Data
D.0	0	0	0	0	0	0	L04	L22' L31' E'	*	0	1	1	0	0	0	–	100111
D.1	1	0	0	0	0	0	L13·E'	L22' L31' E'	*	1	0	0	0	1	0	–	011101
D.3	1	1	0	0	0	0	L22·E'		x	1	1	0	0	0	1	0	
D.7	1	1	1	0	0	0		L31·D'·E'	–	1	1	1	0	0	0	0	000111
D.15	1	1	1	1	0	0	L40	L22' L31' E'	*	1	0	1	0	0	0	–	010111
D/K.23	1	1	1	0	1	x		L22' L13' E	–	1	1	1	0	1	0	*	000101
D.31	1	1	1	1	1	0	L40, L40·E	L22' L13' E	–	1	0	1	0	1	1	*	010100

	Data to Send				Classification			Transmitted Data					
Name	F	G	H	K	Bit Encoding[1]	Disparity[1,2,3]	Din	f	g	h	j	Dout	Alternate Data
D/K.x.0	0	0	0	x	F'·G'·H'	F'·G'	*	0	1	0	0	–	1011
D.x.1	1	0	0	0	(F+G)·H'		x	1	0	0	1	0	
D/K.x.3	1	1	0	x		F·G	–	1	1	0	0	0	0011
D.x.P7	1	1	1	0		F·G, F·G·H	–	1	1	1	0	*	0001
D/K.y.A7	1	1	1	x	F·G·H (S+K)[1]	F·G,F·G·H	–	0	1	1	1	*	1000
K.28.1	1	0	0	1	(F+G)·H'	(F+G)·K	*	1	0	0	1	0	0110
K.28.5	1	0	1	1		(F+G)·K	*	1	0	1	0	0	0101

Table II: 3b/4b Encoding

5. 3b/4b Definition

Table II shows examples of the coding scheme for the 3b/4b module. The 3b/4b follows the same convention and notation as previously shown in Table I for the 5b/6b code. Notice under the Name column in Table II that D.x.1 represents any pattern of ABCDE with the decimal 1 for the FGH data. For the case of D/K.x.3 in Table II, the assertion of K indicates whether the data F through H represents data or control information. Note the FGH bits of the data to send in Table II, represent DATA_IN [5:7] respectively.

6. Special Characters

Special characters are defined as extra signal codes needed beyond the 256 (28) characters already established with K=0. When asserted, the input K character (K=1) recognizes that a special character is being transmitted. Special characters are generally used for transmitting code words such as ABORT, RESET, SHUTDOWN, IDLE, and link diagnostics. In the 8b/10b code, 12 special characters exist as shown below in Table III, where A-H is the data byte to encode, and a-j is the encoded transmitted data. Note, under the Name column K.23.7 represents the decimal 23 in bits A-E and 7 in bits FGH. Also note, the data to send ABCDE FGH is encoded DATA_IN [0:7] respectively. Also in Table 3, Din is the running parity and Dout is the parity of the encoded data. All special characters comply with the general coding constraints of a maximum run length of 5.

Table III: Special Character (K=1)

Name	Data to Send ABCDE FGH K	Din	Transmitted Data abcdei fghj	Dout	Alternate Data
K.28.0	00111 000 1	–	001111 0100	0	110000 1011
K.28.1[1]	00111 100 1	–	001111 1001	*	110000 0110
K.28.2	00111 010 1	–	001111 0101	*	110000 1010
K.28.3	00111 110 1	–	001111 0011	*	110000 1100
K.28.4	00111 001 1	–	001111 0010	0	110000 1101
K.28.5[1]	00111 101 1	–	001111 1010	*	110000 0101
K.28.6	00111 011 1	–	001111 0110	*	110000 1001
K.28.7[1]	00111 111 1	–	001111 1000	0	110000 0111
K.23.7	11101 111 1	–	111010 1000	0	000101 0111
K.27.7	11011 111 1	–	110110 1000	0	001001 0111
K.29.7	10111 111 1	–	101110 1000	0	010001 0111
K.30.7	01111 111 1	–	011110 1000	0	100001 0111

7. Operational Flows of 16b/20b encoder and decoder

The operational flows of a 16b/20b encoder and decoder involve the processes of encoding 16-bit data into 20-bit symbols and then decoding those symbols back into the original data. Figure 3 & 4 shows the flow of encoding and decoding of 16b/20b respectively. Overall, the operational flow of a 16b/20b encoder and decoder is a crucial part of many communication systems, ensuring that data is encoded for reliable transmission and then decoded accurately at the receiving end. The encoding and decoding processes are typically designed to meet the specific requirements and standards of the communication protocol being used.

8. Design Implementation Steps

- Understand the specifications of 16b/20b Encoder/Decoder
- Code the logic with VHDL
- Simulate the VHDL code
- Synthesize the code in Xilinx ISE
- Program the board consisting of Spartan 3AN FPGA
- Test the signals using ChipScope pro analyzer

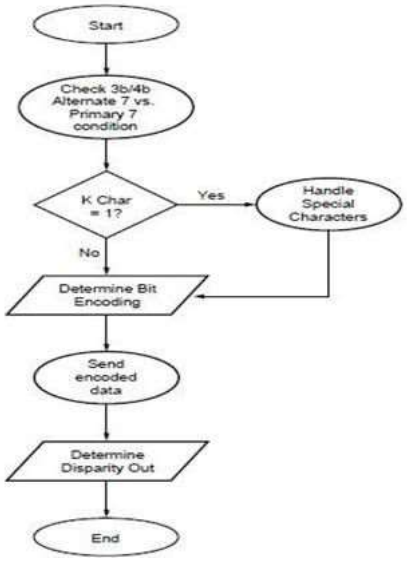

Figure 3: Encoding Flow Diagram

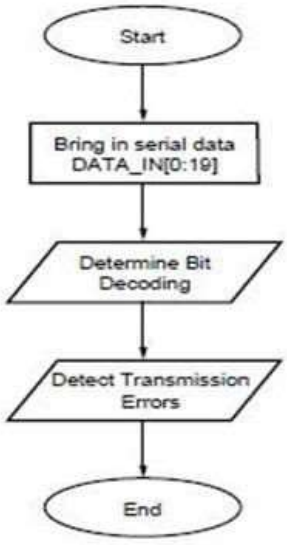

Figure 4: Decoding Flow Diagram

9. Results

The encoder and decoder VHDL designs were validated using Xlinix ISE in Project Navigator simulation. The design has undergone functional and timing model verification after being fitted into a FPGA. Each encoder and decoder module's control, data, and timing were driven by the test bench. This control is merely a model for understanding handshaking while employing a 16b/20b transmission system. Modifications to the control signals used in the source code and test benches given may be required for implementation in an actual system. The test environment supplied in this design instantiates both the encoder and decoder modules and monitors data transmission to/from each module. The output produced by each module during the transmission of the data pattern 00000000 01100011. The ENCODED_DATA is then sent serially as the pattern 1001110100 1011010011. The disparity into the encoder (based on prior data transmission) when sending is negative (dis_in=1), while the running disparity out is positive (dis_out=0). After waiting for some lag, the test bench included with this design verifies the data coming from the decoder. The data is referred in reverse within each lower and upper byte, with the MSB and LSB flipped.

10. Conclusions

This paper has detailed the design of two modules for 8b/10b communication protocol. The 16b/20b encoder design enables 16-bit data transmission serially over any fiber channel using the 8b/10b encoding rules. The 20b/16b decoder module decodes a 20-bit serially encoded stream of data into a 16-bit data word. Both of these designs demonstrate the ability to use Xilinx Spartan 3AN FPGAs for communication protocols when low-power, high-performance, and reliable network devices are required.

REFERENCES

[1]. IEEE: 16B/20B CODEC Development and Its ASIC Implementation; Suma,M.S.; Rekha,S.S.; Future Computer and Communication, 2009. ICFCC 2009. International Conference on 3-5 April 2009 Page(s):622 - 626 Digital Object Identifier 10.1109/ICFCC.2009.23.

[2]. IEEE: A single-chip 3.5 Gb/s CMOS/SIMOX transceiver with automatic-gain-control and automatic-power- control circuits; Ohtomo, Y.; Yoshida, T.; Nishisaka, M.; Nishimura, K.; Shimaya, M.; Solid-State Circuits Conference, 2000. Digest of Technical Papers. ISSCC. 2000 IEEE International 7-9 Feb. 2000 Page(s):58 – 59; Digital Object Identifier 10.1109/ISSCC.2000.839690

[3]. Chandra Shekar P, Design of sequential circuits using adiabatic logic in International Journal of Research Electronics – a peer reviewed bearing a ISSN No: 2349-252X in volume – 5 issue – 2 2018.

[4]. Chandra Shekar P, Design of 64 X 64 single port SRAM Array Using 45nm Technology" in ICRTST-2020 ISSN No. 978-93-5396-830-4 issue 1 vol. 1 page 64

[5]. Halsall, Fred, Data Communications, Computer Networks and Open Systems. Fourth Edition. Addison-Wesley Publishing Company. 1996.

[6]. Widmer, A.X. & Franaszek, P.A., A DC-Balanced, Partitioned-Block, 8b/10b Transmission Code. IBM J. Res. Develop., Vol. 27, No. 5. September 1983.

[7]. Gigabit Ethernet And Fibre Channel Technology. Optimized Engineering Technical Compendium Corporation. (http://www.optimized.com)
[8]. Spartan 3AN Data Sheet: www.xilinx.com
[9]. Modelsim User Guide: www.xilinx.com
[10]. Xilinx ISE User Guide: www.xilinx.com
[11]. ChipScope Pro User Guide: www.xilinx.com
[12]. 8B/10B Encoding Basics: www.wikipedia.com