Key Design Features

- Technology independent soft IP Core for FPGA, ASIC and SoC devices
- Supplied as human readable VHDL (or Verilog) source code
- Separate Encoder/Decoder pair that implements the standard IBM® 8b/10b line code for a DC-balanced serial data stream
- Fully synchronous design with data input and output valid flags
- Encoder and Decoder have a latency of 1 clock cycle
- Generic parallel input and output data widths (specified in whole numbers of input and output bytes)
- Supports all standard control bytes - K.28.0 to K.30.7
- Error flags indicate control byte errors and general decoding errors
- Running Disparity (RD) calculations handled internally and encoder also provides current RD as an output for downstream devices
- Fully scalable, generic architecture
- Suitable for use in serial data links of 6 GHz+ on basic FPGA devices[1]

Applications

- High speed serial interfaces with embedded clocking
- Transmission of DC-balanced, AC-coupled serial data
- Prerequisite for the transmission of serial data over large distances
- Ideal for use in generic serial links (e.g. LVDS or any differential electrical standard)
- Encoding scheme used in a wide range of communication systems such as Ethernet, PCI Express, DVB-ASI, Digital Audio, Fibre Channel, etc.

Generic Parameters

Generic name	*Description*	*Type*	*Valid range*
nb (encoder)	Number of bytes to encode in parallel	integer	≥ 1
nw (decoder)	Number of 10-bit words to decode in parallel	integer	≥ 1

Block Diagram

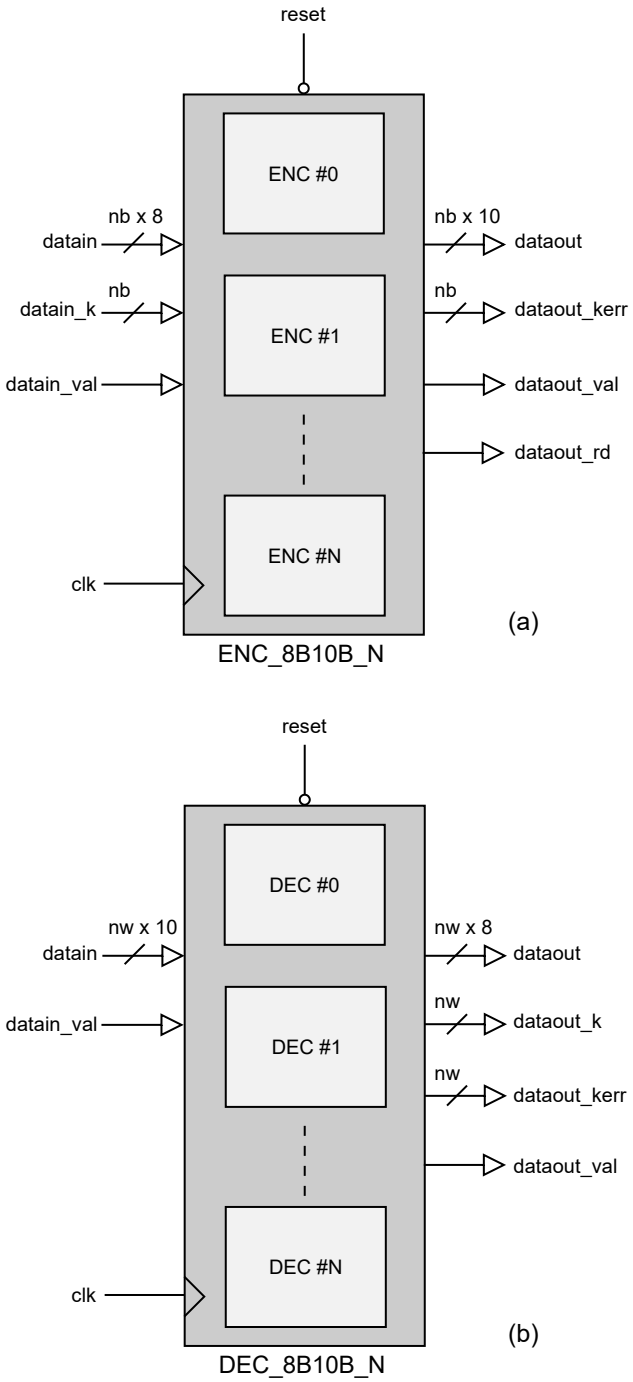

Figure 1: Figure 1: Simplified architectures for the: (a) 8b/10b Encoder and (b) 8b/10b Decoder modules

1 Xilinx® 7-series used as a reference

Pin-out Description

8B10B ENCODER

Pin name	I/O	Description	Active state
clk	in	System clock (Synchronous with parallel input data)	rising edge
reset	in	Asynchronous reset	low
datain [nb * 8 - 1:0]	in	N x input data bytes	data
datain_k [nb - 1:0]	in	N x control byte enables (Indicates that byte is a control symbol)	high
datain_val	in	Input data valid	high
dataout [nb * 10 - 1:0]	out	N x 10-bit output data words	data
dataout_kerr [nb - 1:0]	out	N x control byte error flags (Indicates an unrecognised control symbol input)	high
dataout_rd	out	Output running disparity flag	high: surplus of '1's' low: surplus of '0's'
dataout_val	out	Output data valid	high

8B10B DECODER

Pin name	I/O	Description	Active state
clk	in	System clock (Synchronous with parallel input data)	rising edge
reset	in	Asynchronous reset	low
datain [nw * 10 - 1:0]	in	N x 10-bit input data words	data
datain_val	in	Input data valid	high
dataout [nw * 8 - 1:0]	out	N x output data bytes	data
dataout_k [nw -1:0]	out	N x control byte flags (Indicates that byte is a control symbol)	high
dataout_kerr [nw - 1:0]	out	N x decoding error flags (Indicates an error in the decoded output byte)	high
dataout_val	out	Output data valid	high

General Description

The CODEC_8B10B IP Core is a scalable 8B/10B Encoder/Decoder pair suitable for a wide range of serial data transmission applications. The design is optimized for very high-speed operation and is suitable for use in serial data links of 6 GHz+ on basic FPGA devices.

The design is comprised of an independent encoder and decoder module (Figure 1) that may be used separately or together as a combined CODEC unit. The architecture is fully scalable allowing any number of bytes or words to be encoded or decoded in parallel. The number of bytes and words are specified by the *nb* and *nw* generic parameters.

The encoder and decoder feature a simple synchronous user interface. Data inputs and flags are sampled on the rising-edge of *clk* when the *datain_val* signal is active high. Likewise, data outputs and flags are valid on the rising-edge of *clk* when *dataout_val* is active high.

As well as regular 8-bit data, the encoder can accept one of the special control characters or symbols. These symbols are labelled K.28.0 to K.30.7 and, by convention, are used to insert 'commas' and control characters in a transmitted bitstream. The table below shows the standard symbols supported:

Input	HGFEDCBA	Current RD -ve abcdeifghj	Current RD +ve abcdeifghj
K.28.0	00011100	0011110100	1100001011
K.28.1	00111100	0011111001	1100000110
K.28.2	01011100	0011110101	1100001010
K.28.3	01111100	0011110011	1100001100
K.28.4	10011100	0011110010	1100001101
K.28.5	10111100	0011111010	1100000101
K.28.6	11011100	0011110110	1100001001
K.28.7	11111100	0011111000	1100000111
K.23.7	11110111	1110101000	0001010111
K.27.7	11111011	1101101000	0010010111
K.29.7	11111101	1011101000	0100010111
K.30.7	11111110	0111101000	1000010111

In order to specify that the data input to the encoder is a control symbol then the signal *datain_k* should be set appropriately. Setting this bit high in the respective byte position will make the encoder interpret the input as a control symbol. If the control symbol is not recognized then the encoder will assert the *dataout_kerr* signal with the respective output data.

The decoder has a reciprocal operation to the encoder and decodes the N x 10-bit data back to the original N x bytes. If the decoder detects a decoding error in the stream then the *dataout_kerr* flag will be asserted high for the respective byte.

All Running Disparity (RD) calculations are handled internally in the encoder core. In addition, the current RD is provided as an output flag from the encoder. Resetting the encoder will initialize the current running disparity to zero by default. In addition, the current RD is also provided as an output flag for use by downstream devices. Both the encoder and decoder have a latency of 1 clock cycle from input valid to output valid.

Functional Timing

Figure 2 below shows an example series of three consecutive bytes being encoded. The third byte in the sequence is specified as a control symbol by asserting the *datain_k* flag high.

Inputs are sampled on the rising edge of *clk* when *datain_val* is high. Inputs are ignored otherwise. The encoder has a latency of 1 clock cycle from input to output.

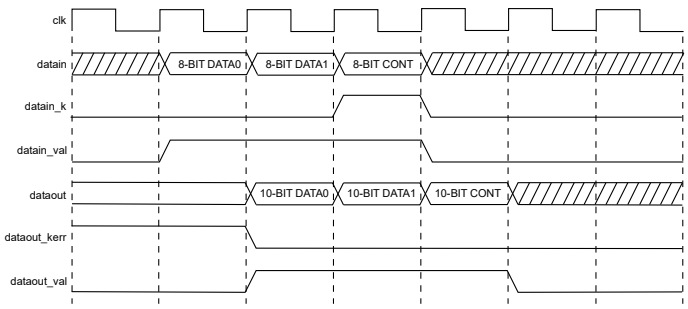

Figure 2: Example 8b/10b encoder timing waveform for a sequence of bytes

A similar waveform is shown in Figure 3 for the decoding process. Again the third word in the sequence is identified as a control symbol with the *dataout_k* flag asserted high. The latency is 1 clock cycle.

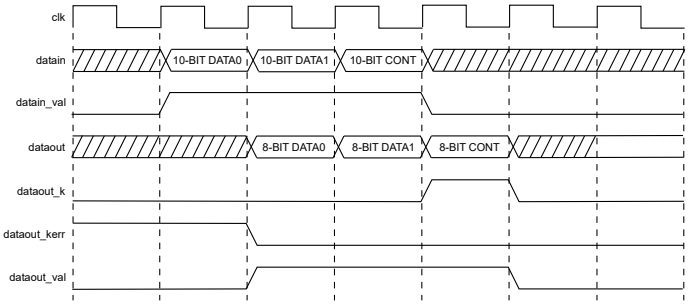

Figure 3: Example 8b/10b decoder timing waveform for a sequence of 10-bit words

Source File Description

All source files are provided as text files coded in VHDL. Equivalent Verilog versions of the code may be provided on request.

Source file	*Description*
enc_8b10b_lut.vhd	8b/10b encoding table
enc_8b10b_code.vhd	8b/10b encoding table (symbols)
enc_8b10b.vhd	Core encoder component
enc_8b10b_n.vhd	Top-level encoder component
dec_8b10b_code.vhd	8b/10b decoding table
dec_8b10b_lut.vhd	8b/10b decoding table (symbols)
dec_8b10b.vhd	Core decoder component
dec_8b10b_n.vhd	Top-level decoder component
codec_8b10b_bench.vhd	Top-level encoder/decoder testbench

Functional Testing

An example testbench is provided for use in a suitable VHDL simulator. The compilation order of the source code is as follows:

1. enc_8b10b_lut.vhd
2. enc_8b10b_code.vhd
3. enc_8b10b.vhd
4. enc_8b10b_n.vhd
5. dec_8b10b_code.vhd
6. dec_8b10b_lut.vhd
7. dec_8b10b.vhd
8. dec_8b10b_n.vhd
9. codec_8b10b_bench.vhd

The testbench instantiates the encoder and decoder IP Core components in series such that a sequence of random 32-bit words are encoded into a a sequence of 40-bit words and vice-versa.

The generic parameters *nb and nw* have been set to 4 to correspond to 4 bytes/words in parallel. The user is free to modify the input and output data widths as required.

The simulation must be run for at least 2 ms during which time the encoder is fed a random sequence of 32-bit data and control symbols. Two output text files are generated during the course of the simulation. These files are 'data_in.txt' and 'data_out.txt' and contain a list of data words captured at the inputs and outputs of the encoder and decoder components. The equivalence of these files proves the correct operation of the test.

Synthesis and Implementation

The files required for synthesis and the design hierarchy is shown below:

- enc_8b10b_n.vhd
  - enc_8b10b.vhd
    - enc_8b10b_lut.vhd
    - enc_8b10b_code.vhd

- dec_8b10b_n.vhd
  - dec_8b10b.vhd
    - dec_8b10b_lut.vhd
    - dec_8b10b_code.vhd

The VHDL core is designed to be technology independent. However, as a benchmark, synthesis results have been provided for the Xilinx® 7-series FPGA devices. Synthesis results for other FPGAs and technologies can be provided on request.

There are no special constraints required for synthesis although reducing the number of parallel encoders and/or decoders in the design will obviously impact the total resource use. The IP Core is completely technology independent.

Trial synthesis results are shown with the generic parameters *nb* = 4 and *nw* = 4. The resource usage is specified after place and route of the design.

ENC_8B10B_N

Resource type	***Artix-7***	***Kintex-7***	***Virtex-7***
Slice Register	46	46	46
Slice LUTs	278	281	281
Block RAM	0	0	0
DSP48	0	0	0
Occupied Slices	107	113	107
Clock freq. (approx)	600 MHz	700 MHz	800 MHz

DEC_8B10B_N

Resource type	***Artix-7***	***Kintex-7***	***Virtex-7***
Slice Register	41	41	41
Slice LUTs	1155	1158	1158
Block RAM	0	0	0
DSP48	0	0	0
Occupied Slices	496	324	498
Clock freq. (approx)	600 MHz	700 MHz	800 MHz

Revision History

Revision	***Change description***	***Date***
1.0	Initial revision	01/12/2014
1.1	Updated synthesis results for Xilinx® 7 series devices	18/03/2015
1.2	Included RD flag as an output port in the encoder to make available for downstream devices	01/09/2020

X-ON Electronics

Largest Supplier of Electrical and Electronic Components

Click to view similar products for Development Software ***category:***

Click to view products by Zipcores ***manufacturer:***

Other Similar products are found below :

SRP004001-01 SW163052 SYSWINEV21 WS01NCTF1E W128E13 SW89CN0-ZCC IP-UART-16550 MPROG-PRO535E AFLCF-08-LX-CE060-R21 WS02-CFSC1-EV3-UP SYSMAC-STUDIO-EIPCPLR 1120270005 SW006021-2H ATATMELSTUDIO 2400573 2702579 2988609 SW006022-DGL 2400303 88970111 DG-ACC-NET-CD 55195101-101 55195101-102 SW1A-W1C MDK-ARM SW006021-2NH B10443 SW006021-1H SW006021-2 SW006022-2 SW006023-2 SW007023 MIKROE-730 MIKROE-2401 MIKROE-499 MIKROE-722 MIKROE-724 MIKROE-726 MIKROE-728 MIKROE-732 MIKROE-734 MIKROE-736 MIKROE-738 MIKROE-744 MIKROE-928 MIKROE-936 1120270002 1120270003 1120275015 NT-ZJCAT1-EV4