8b10b Encoder/Decoder MegaCore Function (ED8B10B)

November 2001; ver. 1.02 Data Sheet

Introduction

Encoders and decoders are used for physical layer coding for Gigabit Ethernet, Fibre Channel, and other applications. The 8b/10b encoder takes byte inputs, and generates a direct current (DC) balanced stream (equal number of 1s and 0s) with a maximum run length of 5. Some of the individual 10-bit codes will have an equal number of 1s and 0s, while others will have either four 1s and six 0s, or, six 1s and four 0s. In the latter case, the disparity between 1s and 0s is used as an input to the next 10-bit code generation, so that the disparity can be reversed, and maintain an overall balanced stream. For this reason, some 8-bit inputs have two valid 10-bit codes, depending on the input disparity.

The Altera® 8b10b Encoder/Decoder MegaCore® Function (ED8B10B) is a compact, high performance core capable of encoding and decoding at Gigabit Ethernet rates (125 MHz: 1 Gbps). The ED8B10B is optimized for the APEX™ 20K, FLEX 10K®, and Mercury™ devices.

Features

- Look-up table (LUT)-based implementation of encoder
- Industry compatible special character coding
- Complies with all applicable standards, including:
  - Institute of Electrical and Electronics Engineers, IEEE 802.3z, *Media Access Control (MAC) Parameters, Physical Layer, Repeater and Management Parameters for 1000 Mb/s Operation*, 1998, paragraphs 36.2.4.1 to 36.2.4.6.
- Quartus® II software, and OpenCore® feature allow place-and-route and static timing analysis of designs prior to licensing
- Secure register transfer level (RTL) simulation models allow simulation with user design in third-party simulators

Functional Description

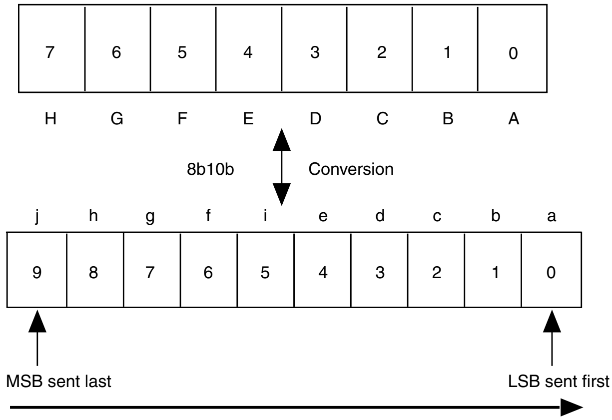
The ED8B10B can encode one 8-bit byte of data into a 10-bit transmission code, and decode a 10-bit code into one 8-bit byte of data. Figure 1 illustrates the bidirectional conversion process.

The eight input bits are named `A`, `B`, `C`, `D`, `E`, `F`, `G`, `H`. Bit `A` is the least significant bit (LSB), and bit `H` is the most significant bit (MSB). They are split into two groups: The five-bit group `A`, `B`, `C`, `D`, `E`, and the three-bit group `F`, `G`, `H`.

The coded bits are named `a`, `b`, `c`, `d`, `e`, `i`, `f`, `g`, `h`, `j` (the order is not alphabetical). These bits are also split into two groups: the six-bit group `a`, `b`, `c`, `d`, `e`, `i`, and the four-bit group `f`, `g`, `h`, `j`.

A-DS-IPED8B10B-1.02

Figure 1. 8b10b Conversion



In bit serial transmission, the LSB is usually transmitted first, while the MSB is usually transmitted last.

Character Codes

In addition to 256 data characters, the 8b/10b code defines twelve out-of-band indicators, also called special control characters. The 256 data characters are named D*x*.*y*, and the special control characters are named K*x*.*y*. The *x* value corresponds to the five-bit group, and the *y* value to the three-bit group.

The special control characters indicate, for example, whether the data is idle, test data, or data delimiters. In applications where encoded characters are transmitted bit-serially, the comma character (K28.5) is usually used for alignment purposes as its 10-bit code is guaranteed not to occur elsewhere in the encoded bit stream, except after K28.7 which is normally only sent during diagnostic.

Table 1 lists the special *K* codes used by the ED8B10B.

Table 1. Character Codes

10-Bit Special *K* Codes	Equivalent 8-Bit Codes
K28.0	8'b000_11100
K28.1	8'b001_11100
K28.2	8'b010_11100
K28.3	8'b011_11100
K28.4	8'b100_11100
K28.5 *(1)*	8'b101_11100
K28.6	8'b110_11100
K28.7	8'b111_11100
K23.7	8'b111_10111
K27.7	8'b111_11011
K29.7	8'b111_11101
K30.7	8'b111_11110

Note:
(1) K28.5 is a comma character used for alignment purposes, and to represent the IDLE code.

Disparity

Disparity is the difference between the number of 1s and 0s in the encoded word.

- Neutral disparity indicates the number of 1s and 0s are equal.
- Positive disparity indicates more 1s than 0s.
- Negative disparity indicates more 0s than 1s.

The ED8B10B is designed to maintain a neutral average disparity. Average disparity determines the DC component of a serial line. Running disparity is a record of the cumulative disparity of every encoded word, and is tracked by the encoder. To guarantee neutral average disparity, a positive running disparity must be followed by neutral or negative disparity; a negative running disparity must be followed by neutral or positive disparity. If these conditions are not met, the decoder flags an error by asserting its `rderr` output.

For details on running disparity rules, see the IEEE 802.3z specification, paragraph 36.2.4.4.

Encoder

The encoder uses an innovative, proprietary, non-partitioned, memory-based LUT implementation to convert data and identified special 8-bit codes from 8 bits to 10 bits. See Figure 1 on page 2 for an illustration of the conversion process.

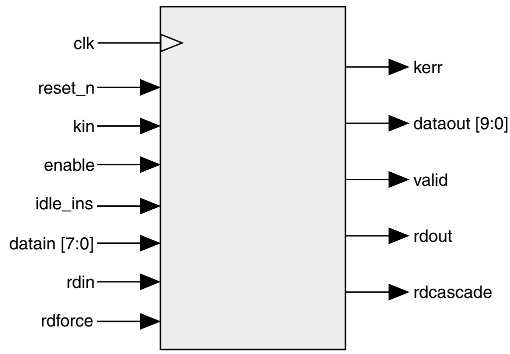
To encode an 8-bit word, the 8-bit value must be applied to the `datain` inputs and the `enable` input must be asserted (active high).

When one of the twelve special 10-bit codes is to be inserted, the equivalent 8-bit code is placed on the `datain` lines and the `kin` input is asserted. The core performs error checking to ensure the out-of-band 8-bit code is valid. If not, the `kerr` output is asserted. See Table 1 on page 3 for a list of the valid *K* codes.

Idle (K28.5) characters can be automatically inserted when `enable` is not asserted by asserting the `idle_ins` input.

Figure 2 shows a block diagram of the encoder.

Figure 2. ED8B10B Encoder



Disparity

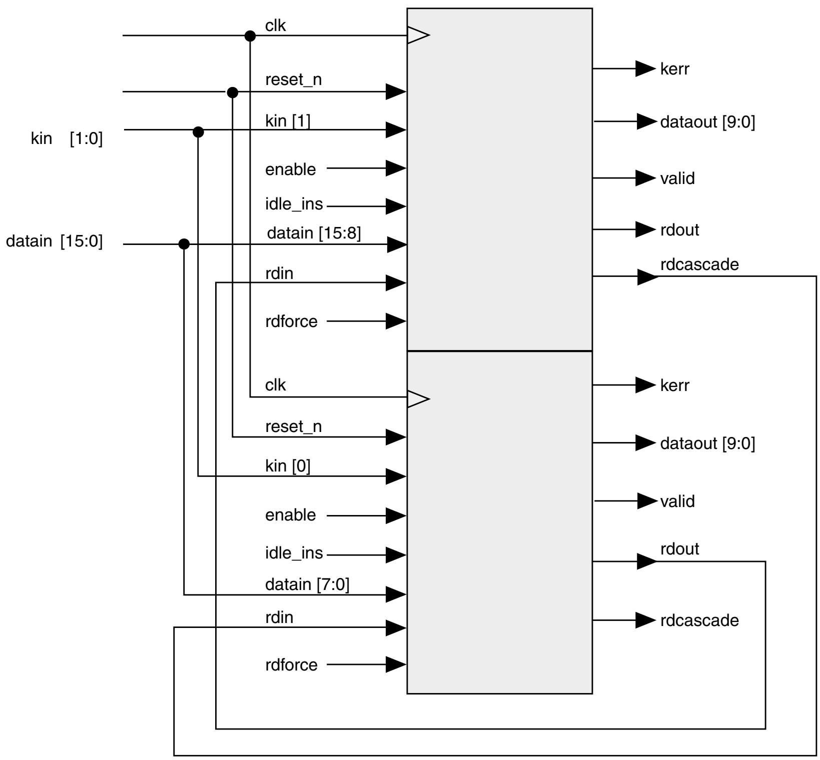
The running disparity can be forced to positive or negative, allowing the user to insert a special resynchronization pattern, or disparity errors.

When the `rdforce` input is asserted, the value on the `rdin` port is assumed to be the current running disparity. Setting `rdin` to 0 forces the encoder to produce an encoded word with positive or neutral disparity. Setting `rdin` to 1 forces the encoder to produce an encoded word with negative or neutral disparity.

Cascaded Encoding

Two encoders can be cascaded to allow for 16-bit word encoding. The encoders are cascaded by connecting the `rdcascade` output of the most significant byte (MSByte) encoder to the `rdin` input of the least significant byte (LSByte) encoder, and by connecting the `rdout` output of the LSByte encoder to the `rdin` input of the MSByte encoder. These connections ensure proper running disparity computation. The `rdforce` inputs must be asserted (active high) for the encoders to take into account the value on the `rdin` inputs, rather than use their internally generated running disparity. Both `enable` inputs must be high or low at the same time. The `kin [1]` signal relates to `datain[15:8]`, and `kin[0]` relates to `datain[7:0]`. Figure 3 shows two encoders connected together to perform cascaded encoding.

Figure 3. Cascaded Encoding *Note (1)*



Note:

(1) The `enable`, `idle_ins`, and `rdforce` signals are set high (logic 1).

Encoding Latency

The encoder is pipelined, thus it takes three clock cycles for a character to be encoded. The encoded value—corresponding to the values of `datain` and `kin` sampled by the encoder on rising edge `n`—is output shortly after rising edge `n+2`, and is available to be sampled on the rising edge of clock cycle `n+3`. (See Figure 4 on page 7).

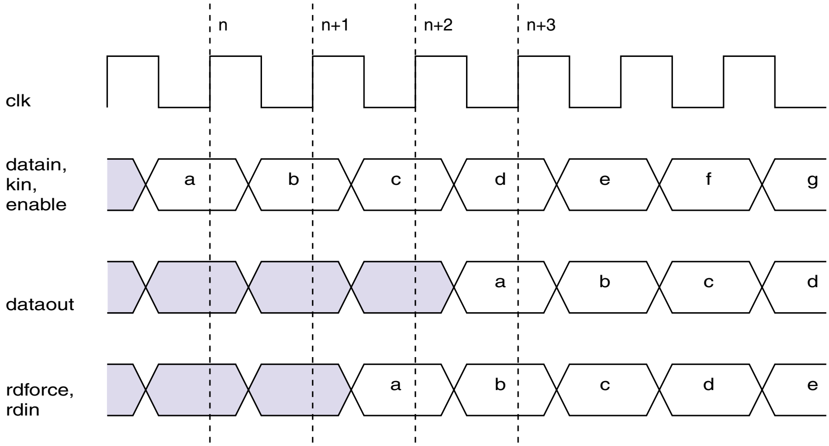
To enable cascaded encoding, the data paths fed by the `rdforce` and `rdin` inputs are not pipelined. Since `rdforce` and `rdin` are normally only used in cascaded configurations, this should not be a problem.

In cases where the `rdforce` and `rdin` inputs are to be used in non-cascaded configurations, they should be delayed two clock cycles with respect to their corresponding `datain` and `kin` values. This can be achieved by inserting two registers in series with each of the inputs to be delayed, the following example Verilog code shows how to implement the required delay registers.

Example: Adding delay to `rdforce` and `rdin` for non-cascaded applications:

```
// The _pre2 registers are set at the same time as datain and kin.
reg rdforce_pre2;
reg rdin_pre2;
// The _pre1 registers provide an extra clock tick of delay
reg rdforce_pre1;
reg rdin_pre1;
always @ (posedge clk) begin
   rdforce <= rdforce_pre1;
   rdforce_pre1 <= rdforce_pre2;
   rdin <= rdin_pre1;
   rdin_pre1 <= rdin_pre2;
end
```

Figure 4. Encoder Timing Diagram



Decoder

Data and identified 10-bit special *K* codes are converted from 10 bits to 8 bits; see Table 1 on page 3 for a list of the valid *K* codes, and Figure 1 on page 2 for an illustration of the conversion process.

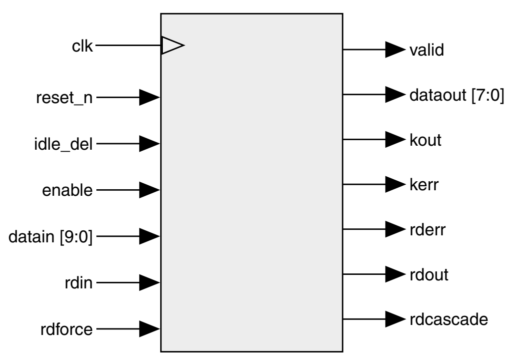
When special 10-bit *K* codes are received, the special *K* codes are translated to 8-bit values, and the `kout` signal is asserted. The decoder also checks for invalid 10-bit codes, and asserts the `kerr` signal when invalid codes are detected.

When the `idle_del` signal is asserted, it deletes all 10-bit words identified as the special IDLE character of K28.5.

When the receiver detects a disparity error, the `rderr` signal is asserted.

Figure 5 shows a block diagram of the decoder.

Figure 5. ED8B10B Decoder



CASCADED DECODING

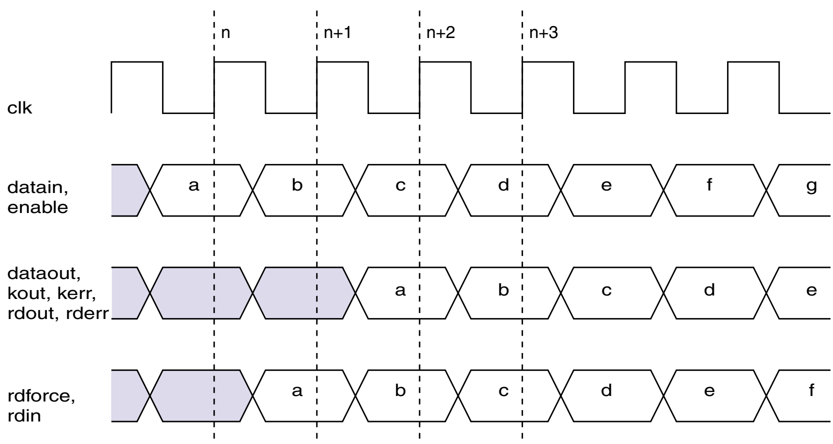
Two decoders can be cascaded to decode two words simultaneously. The decoders are cascaded—in a similar fashion as the encoders—by connecting the `rdcascade` output of the first decoder to the `rdin` input of the second decoder, and by connecting the `rdout` output of the second decoder to the `rdin` input of the first decoder. The `rdforce` inputs of both decoders must be tied high.

To enable cascaded decoding, the data paths fed by the `rdin` and `rdforce` inputs are not cascaded. If these inputs are to be used in non-cascaded decoders, they should be delayed by one clock cycle with respect to their corresponding `datain` and `kin` inputs.

Decoding Latency

The decoder is pipelined, thus it takes two clock cycles for a character to be decoded. The decoded value—corresponding to the value of `datain` sampled by the decoder on rising edge `n`—is output shortly after rising edge `n+1`, and is available to be sampled on the rising edge of clock cycle `n+2`. (See Figure 4 on page 7).

Figure 6. Decoder Timing Diagram



I/O Signals

Tables 2 and 3 list the input/output signals for the encoder, and decoder.

Table 2. Encoder I/O Signals

Signal Name	Direction	Description
clk	Input	Clock. The input is latched, and the result is output on this clock. There is a three clock cycle latency between the input and output.
reset_n	Input	Active low, reset. Asynchronously resets all registers in the core.
kin	Input	Command byte indicator. When high, indicates that the input is a command byte, not a data byte.
enable	Input	Enable encoder signal. When high, indicates that the data currently present on the datain input is to be encoded.
idle_ins	Input	Idle character insert. When high, idle (K28.5) characters are inserted when enable is not asserted.
datain[7:0]	Input	Data input. This is the 8-bit input word, data or command.
rdin	Input	Running disparity input. When rdforce is high, the value on this pin is used as the current running disparity instead of the internally generated one.
rdforce	Input	Force running disparity. When high, the rdin value overrides the internally generated running disparity.
kerr	Output	Special *K* character error. This signal is set high when enable and kin are high and the value on datain is not a valid special *K* character.
dataout[9:0]	Output	Data output. This is the 10-bit encoded output.
valid	Output	Valid signal. When high, indicates that a valid encoded word is present on the dataout output.
rdout	Output	Running disparity output. The current running disparity (after encoding the word present on the dataout output).
rdcascade	Output	Cascaded Running disparity. Used when encoders are cascaded.

Table 3. Decoder I/O Signals

Signal Name	Direction	Description
clk	Input	Clock. The input is latched, and the result output on this clock. There is a three clock cycle latency between the input and output.
reset_n	Input	Active low, reset. Asynchronously resets all registers in the core.
idle_del	Input	Idle delete signal. When high, idle words (K28.5) are removed from the stream (i.e. valid is set low when idle words are received).
enable	Input	Enable decoder signal. When high, indicates that the data currently present on the datain input is to be decoded.
datain[9:0]	Input	Data input. This is the 10-bit encoded input word.

Table 3. Decoder I/O Signals

Signal Name	Direction	Description
`rdin`	Input	Running disparity input. When `rdforce` is high, the value on this pin is used as the current running disparity instead of the internally generated one.
`rdforce`	Input	Force running disparity. When high, the `rdin` value overrides the internally generated running disparity.
`valid`	Output	Valid signal. When high, indicates that a valid decoded word is present on the `dataout` output.
`dataout[7:0]`	Output	Data output. This is the 8-bit decoded data or command.
`kout`	Output	Command output. When high, indicates that the output is a command byte, not a data byte.
`kerr`	Output	Special *K* error. Asserted high when an invalid 10-bit word is received.
`rderr`	Output	Running disparity error. When high indicates the running disparity rules have been violated.
`rdout`	Output	Running disparity output. The current running disparity (after decoding the word present on the `dataout` output).
`rdcascade`	Output	Cascaded Running disparity. Used when decoders are cascaded.

Resources

Table 4 shows the required speed and estimated gate count of the ED8B10B in its possible target devices.

Table 4. Resources Note (1)

Device	Mode	LEs/LCs	ESBs/EABs	f_{MAX} (MHz)
APEX 20K Family EP20K30ETC144-1	Encoder Decoder	56 133	2 0	173 *(2)* 150
Mercury Family EP1M120F484C7AES	Encoder Decoder	56 137	1 0	216 *(2)* 223
Flex 10K Family EPF10K30ETC144-1	Encoder Decoder	58 129	2 0	138 *(2)* 147

Note:

(1) The numbers for the logic elements/logic cells (LEs/LCs) and embedded system blocks/embedded array blocks (ESBs/EABs) are approximate as November 2001.

(2) f_{MAX} is for non-cascaded encoder/decoder.

Licensing

A license is not required to perform the following trial operations using your own custom logic:

- Instantiation
- Place-and-route
- Static timing analysis
- Simulation on a third-party simulator

When you are ready to generate programming files, you need to obtain licenses through your local Altera sales representative.

All current 8b10b Encoder/Decoder MegaCore functions use a single license with ordering code: IP–ED8B10B.

Deliverables

The following elements are provided with the package:

- Data sheet
- Encrypted gate level netlist & **ROM** file
- Place-and-route constraints (where necessary)
- Secure RTL simulation model
- Demo testbench
- Access to problem reporting system

Refer to the **Readme** file included with the package for installation and customization instructions.

ALTERA®
101 Innovation Drive
San Jose, CA 95134
(408) 544-7000
http://www.altera.com
Applications Hotline:
(800) 800-EPLD
Customer Marketing:
(408) 544-7104
Literature Services:
lit_req@altera.com

Copyright © 2001 Altera Corporation. Altera, The Programmable Solutions Company, the stylized Altera logo, specific device designations, and all other words and logos that are identified as trademarks and/or service marks are, unless noted otherwise, the trademarks and service marks of Altera Corporation in the U.S. and other countries. All other product or service names are the property of their respective holders. Altera products are protected under numerous U.S. and foreign patents and pending applications, maskwork rights, and copyrights. Altera warrants performance of its semiconductor products to current specifications in accordance with Altera's standard warranty, but reserves the right to make changes to any products and services at any time 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.

nsai
I.S. EN ISO 9001