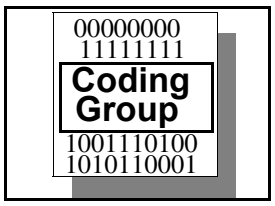
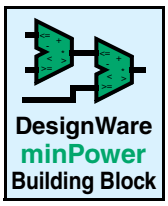




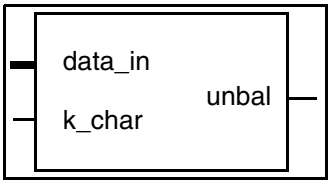
DW_8b10b_unbal

8b10b Coding Balance Predictor

Version, STAR and Download Information: IP Directory

Features and Benefits

- Independent of Running Disparity
- Higher speed than a full encoder
- Predicts balance for both data and special characters



Description

DW_8b10b_unbal predicts (without fully encoding) whether a given character will or will not flip the running disparity of the 8b/10b encoder when it is encoded. This function is useful for implementing multi-byte encoder schemes with some protocols that tightly control the state of the running disparity in the encoder. DW_8b10b_unbal may also be useful even in single byte encoder designs that require pipelining as well as control of the running disparity.

Table 1-1 Pin Description

Pin Name	Width	Direction	Function
k_char	1 bit	Input	Special character control input (LOW for data characters, HIGH for special characters)
data_in	8 bits	Input	Input for 8-bit data character to be encoded
unbal	1 bit	Output	Unbalanced code character indicator (LOW for balanced, HIGH for unbalanced)

Table 1-2 Parameter Description

Parameter	Values	Description
k28_5_mode	0 or 1 Default: 0	Special Character subset control parameter 0 for all special characters available, 1 for only K28.5 available [when `k_char` = HIGH, regardless of the value on `data_in`]

Table 1-3 Synthesis Implementations

Implementation Name	Function	License Feature Required
rtl	Synthesis model	DesignWare

Related Topics

- Coding Group – Coding Overview
- DesignWare Building Block IP Documentation Overview

HDL Usage Through Component Instantiation - VHDL

```
library IEEE,DWARE,DWARE;
use IEEE.std_logic_1164.all;
use DWARE.DW_foundation_comp.all;

entity DW_8b10b_unbal_inst is
  generic (inst_k28_5_only : integer := 0 );
  port (inst_k_char  : in std_logic;
        inst_data_in : in std_logic_vector(7 downto 0);
        unbal_inst   : out std_logic );
end DW_8b10b_unbal_inst;

architecture inst of DW_8b10b_unbal_inst is
begin

  -- Instance of DW_8b10b_unbal
  U1 : DW_8b10b_unbal
    generic map (k28_5_only => inst_k28_5_only )
    port map (k_char => inst_k_char,   data_in => inst_data_in,
              unbal => unbal_inst );
end inst;

-- Configuration for use with VSS simulator
-- pragma translate_off
configuration DW_8b10b_unbal_inst_cfg_inst of DW_8b10b_unbal_inst is
  for inst
  end for; -- inst
end DW_8b10b_unbal_inst_cfg_inst;
-- pragma translate_on
```

HDL Usage Through Component Instantiation - Verilog

```
module DW_8b10b_unbal_inst( inst_k_char, inst_data_in, unbal_inst );

  parameter k28_5_only = 0;

  input inst_k_char;
  input [7 : 0] inst_data_in;
  output unbal_inst;

  // Instance of DW_8b10b_unbal
  DW_8b10b_unbal #(k28_5_only)
    U1 (.k_char(inst_k_char),  .data_in(inst_data_in),  .unbal(unbal_inst) );
endmodule
```

Copyright Notice and Proprietary Information

Destination Control Statement

All technical data contained in this publication is subject to the export control laws of the United States of America. Disclosure to nationals of other countries contrary to United States law is prohibited. It is the reader's responsibility to determine the applicable regulations and to comply with them.

Disclaimer

SYNOPSYS, INC., AND ITS LICENSORS MAKE NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

Trademarks

Synopsys and certain Synopsys product names are trademarks of Synopsys, as set forth at https://www.synopsys.com/company/legal/trademarks-brands.html.

All other product or company names may be trademarks of their respective owners.

Third-Party Links

Any links to third-party websites included in this document are for your convenience only. Synopsys does not endorse and is not responsible for such websites and their practices, including privacy practices, availability, and content.

Synopsys, Inc.
690 E. Middlefield Road
Mountain View, CA 94043

www.synopsys.com