

Compa Family LVDS4to1 and LVDS8to1 Application Guide

(AN03016, V1.0)

(24.08.2021)

Shenzhen Pango Microsystems Co., Ltd.

All Rights Reserved. Any infringement will be subject to legal action.

Revisions History

Document Revisions

Version	Date of Release	Revisions
V1.0	24.08.2021	Initial release

Application Examples For Reference Only

About this Manual

Terms and Abbreviations

Terms and Abbreviations	Meaning
LVDS	Low-Voltage Differential Signaling

Application Examples For Reference Only

Table of Contents

Revisions History	1
About this Manual	2
Table of Contents	3
Tables	4
Figures	5
Chapter 1 Overview	6
1.1 Introduction	6
1.1.1 Main Functions	6
1.1.2 Resource Usage	6
Chapter 2 Function Description Transmitter	7
2.1 Transmitter Logic Block Diagram	7
2.1.2 Transmitter Interface Definition	7
2.2 Receiver	8
2.2.1 Receiver Logic Block Diagram	8
2.2.2 Receiver Interface Definition	10
Chapter 3 Reference Designs	11
3.1 Reference Design Project Directory	11
3.1.1 Transmitter Reference Design Project Directory	11
3.1.2 Receiver Reference Design Project Directory	12
3.2 Reference Design Board Validation	13
Disclaimer	14

Tables

Table 1-1 Resource Usage	6
Table 2-1 Transmitter Interface Definition	7
Table 2-2 Receiver Interface Definition	10

Application Examples For Reference Only

Figures

Figure 2-1 Transmitter Logic Block Diagram	7
Figure 2-2 Receiver Logic Block Diagram.....	8
Figure 3-1 Reference Design Project Directory.....	11
Figure 3-2 PLL Parameter Configuration	12
Figure 3-3 Reference Design Project Directory.....	12
Figure 3-4 Verification Hardware Environment Block Diagram.....	13

Chapter 1 Overview

1.1 Introduction

This document serves as a system-level application scheme for the Compa family LVDS4to1 and LVDS8to1 launched by Shenzhen Pango Microsystems Co., Ltd. It mainly introduces the function list, design architecture, interface definition, and reference designs of LVDS4to1 and LVDS8to1.

1.1.1 Main Functions

The main functions supported are:

- 4:1, 8:1 serialization of data, transmission of synchronous clocks; reception of serialized data and 1:4, 1:8 deserialization;
- Automatic phase alignment and word boundary alignment functions;
- The number of channels can be parameterized for expansion.

1.1.2 Resource Usage

The LVDS4to1 and LVDS8to1 reference design scheme has 1 clock lane with 4 data lanes, with the main resource usage as shown in [Table 1-1](#).

Table 1-1 Resource Usage

	FF	LUT	IO	IOCLKGATE	PLL	UCMDC	CLKDIV	CCS
LVDS4:1 Reception	109	136	16	1	1	3	1	1
LVDS4:1 Transmission	379	401	20	1	0	2	1	1
LVDS8:1 Reception	137	188	16	1	1	3	1	1
LVDS8:1 Transmission	513	567	20	1	0	2	1	1

Chapter 2 Function Description Transmitter

2.1 Transmitter Logic Block Diagram

The logic at transmitter mainly completes the 4:1 and 8:1 serial-to-parallel conversion of data to be sent, synchronous clock output, etc., and the LVDS driver function is implemented by the bottom-level unit GTP_OUTBUFDS of the Compa family devices. The block diagram is presented in Figure 2-1.

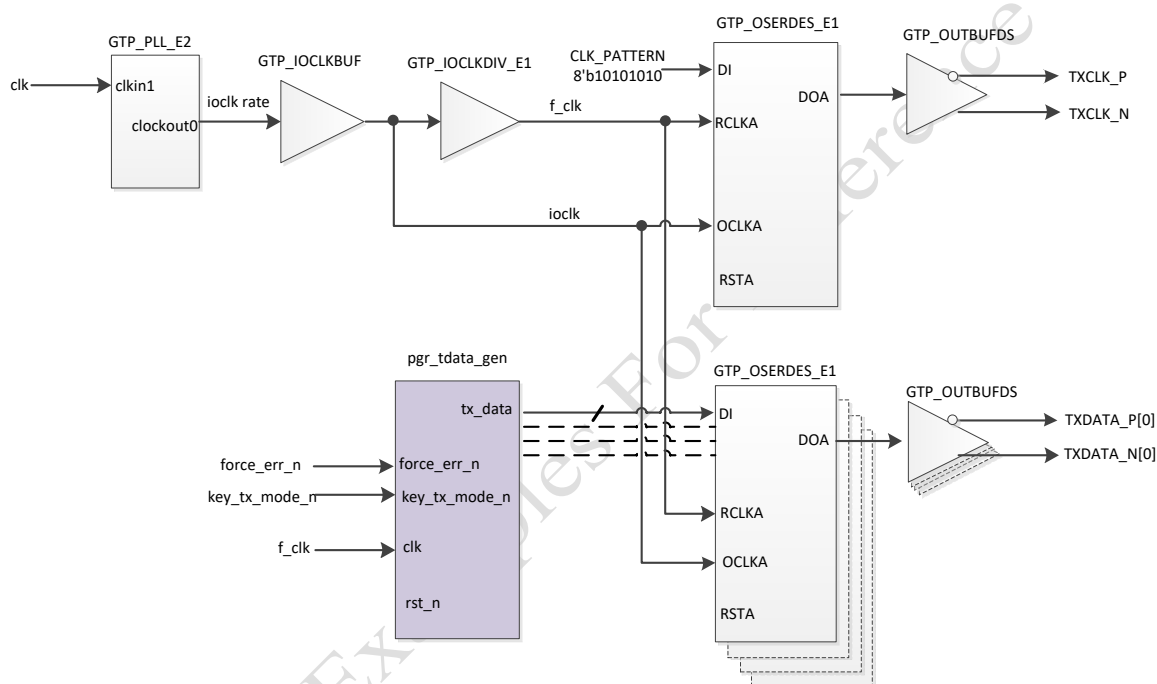


Figure 2-1 Transmitter Logic Block Diagram

Herein, the synchronous clock is obtained by serializing CLK_PATTERN=8'b10101010 at 4:1 or 8:1 through GTP_OSERDES_E1. The pgr_tdata_gen module is used to generate parallel test data, and can switch output between four types of test data via the key_tx_mode_n signal, namely CLK_PATTERN type, word boundary alignment type, CNT type, and PRBS-7 type data.

2.1.2 Transmitter Interface Definition

Table 2-1 Transmitter Interface Definition

Signals	Direction	Bit width	Description
clk	Input	1	Transmitter reference clock (50MHz)
rst_n	Input	1	Transmitter reset signal, active-low
key_tx_mode_n	Input	1	Connected to an external button, active-low, used to switch the mode of the transmit data.

Signals	Direction	Bit width	Description
force_err_n	Input	1	Connected to an external button, active-low, each press inserts a 1-bit error into the transmit data.
tx_mode	Output	2	Transmitter data type display
tx_data_p/n	Output	CHAN_N	Transmitter data differential signal
tx_clk_p/n	Output	1	Transmitter clock differential signal

2.2 Receiver

2.2.1 Receiver Logic Block Diagram

The receiver logic mainly completes synchronous clock recovery, 1:4 or 1:8 serial-to-parallel conversion, clock data phase alignment, word boundary alignment, bit error testing, and other tasks. The block diagram is shown in the following figure:

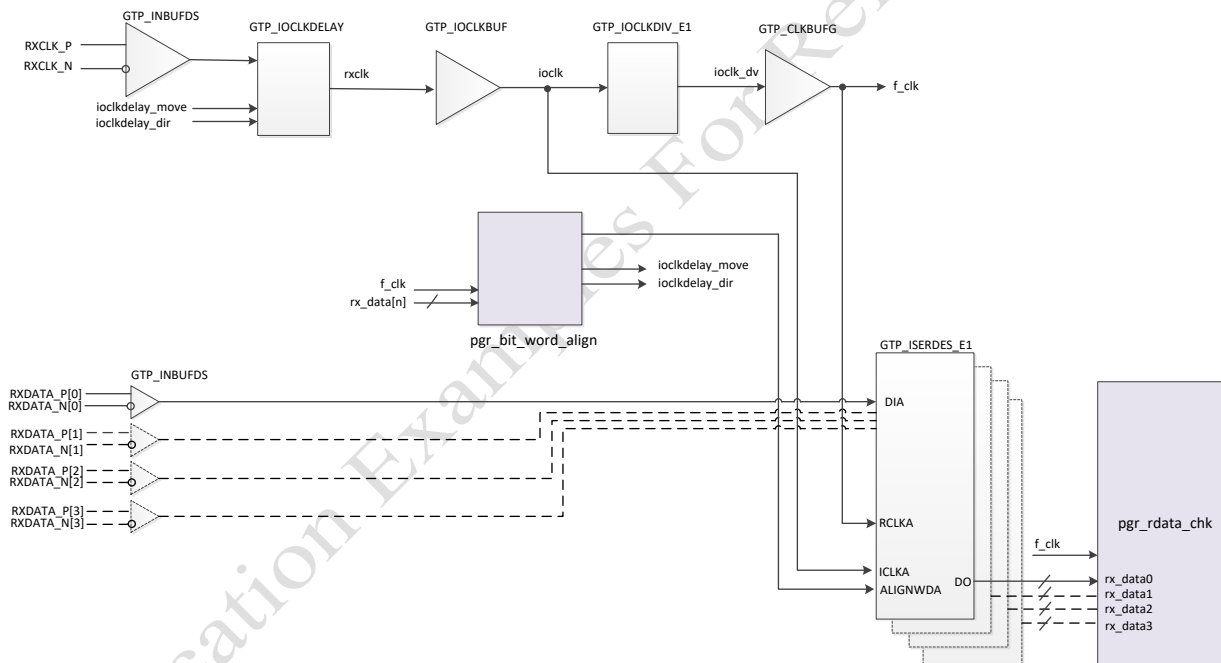


Figure 2-2 Receiver Logic Block Diagram

➤ Synchronous Clock Recovery

The synchronous clock is recovered to a high-speed serial clock ioclk and a low-speed parallel clock f_clk that is phase-aligned with the serial clock, both of which are used for data sampling. Implemented by the underlying hardware units GTP_IOCLKDIV_E1, GTP_IOCLKBUF, and GTP_CLKBUFG of the PGC device.

Wherein, GTP_IOCLKBUF is an IO clock buffer used to connect the delayed clock to the IO clock network; GTP_IOCLKDIV_E1 is an IO clock divider, used for dividing the serial clock

by a factor of 2 (1:4) or 4 (1:8) to obtain a parallel clock; GTP_CLKBUFG is a global clock buffer, used to connect the parallel clock to the global clock network.

➤ 1:4 or 1:8 Serial-to-Parallel Conversion

Samples serial data and outputs it in parallel as a word of 4 bits or 8 bits. Implemented by the underlying unit GTP_ISERDES_E1 of the PGC device.

➤ Clock Data Phase Alignment

Aligns the edges of the recovered serial clock ioclk to the center of the serial data sampling window. Implemented by the underlying unit GTP_IOCLKDELAY of the PGC device. The GTP_IOCLKDELAY can statically or dynamically adjust the phase of the clock.

In this scheme, the relative phase adjustment between the sampling clock and data is implemented by adjusting the delay value of GTP_IOCLKDELAY. The number of delay adjustment steps for GTP_IOCLKDELAY is 127, with each step being approximately 15ps. For ease of expression, each delay value from level 0 to 127 is represented by tapx, for example, tap64 represents the 64th level of delay.

The purpose of phase adjustment is to find a continuous tap range, such as tap8~tap32, within which data can be sampled stably. This tap range is referred to as the sampling window. After finding the sampling window, move the DELAY of GTP_IOCLKDELAY to the center tap position of the maximum sampling window, this process is called clock data phase alignment. If the path delay from the PAD of the Compa family device to the GTP_ISERDES_E1 data input port DIA inside the CPLD is completely consistent for the synchronous clock and each channel's data (or the skew is negligible), it can be assumed that the sampling windows of the synchronous clock and data are completely aligned. By scanning the sampling window of the synchronous clock, the sampling window of the data can be determined.

In cases where the skew between the synchronous clock, data, and data channels is negligible, the optional module pgr_bit_word_align can achieve automatic phase alignment. After reset, the module triggers the GTP_IOCLKDELAY unit to move the ioclk by one tap delay every $2^{20}=1048576$ f_clk cycles through the MOVE signal, until it stops after the 127th tap.

During each tap duration, the module continuously judges the 1,048,576 (i.e., 2^{20}) deserialized values of clk_s2p. If these deserialized values are all equal to the same value (for example, all are 4'b1010) and rx_data[n] belong to {4'b1010, 4'b0101}, then the current tap is marked as a valid tap, otherwise, it is marked as invalid. A sequence of valid taps forms a sampling window.

Between 0 to 127 taps, there may be one or more sampling windows. The pgr_bit_word_align module records the starting position and width of the largest sampling window during traversal,

and automatically adjusts the DELAY of GTP_IOCLKDELAY to the center tap position of the largest sampling window upon completion of the traversal, thereby completing the automatic alignment of the clock data phase.

When there is significant skew between the synchronous clock, data, and data channels, it is recommended to use GTP_IOCLKDELAY and GTP_IODELAY_E1 static delay compensation to complete the alignment of the clock data phase.

➤ Word Boundary Alignment

After the alignment of the clock data phase is completed, if a word boundary alignment command is received, the soft logic can automatically complete the word boundary alignment.

➤ Bit Error Testing

Once the clock data phase alignment and word boundary alignment are finished, the pgr_rdata_chk module on the receiving end begins to determine whether the received data contains errors; if so, it outputs the corresponding indicator signal.

2.2.2 Receiver Interface Definition

Table 2-2 Receiver Interface Definition

Signals	Direction	Bit width	Description
clk	Input	1	Reference clock (50MHz)
rst_n	Input	1	Reset signal, active low
rx_data_p/n	Input	CHAN_N	Receiver data differential signal
rx_clk_p/n	Input	1	Receiver clock differential signal
dbg_key_oper	Output	1	Manually adjust ioclkdelay step, constrained to a key press
dbg_key_mode	Output	1	Adjustment mode selection, constrained to a key press, mode=0 corresponds to step adjustment, mode=1 corresponds to operation direction, mode=2 corresponds to word boundary alignment
dbg_rd_data	Output	1	Single-bit data read from the RAM module
f_clk_dv2	Output	1	The system's division clock is further divided by two for output
word_align_done	Output	1	Status signal, output high indicates word boundary alignment is complete
flash_led	Output	1	Input heartbeat signal
BERT_result_flag	Output	1	Data reception error indication, output high indicates the reception of errors
dbg_c_tap	Output	1	Indication that ioclkdelay step is non-zero

Chapter 3 Reference Designs

3.1 Reference Design Project Directory

3.1.1 Transmitter Reference Design Project Directory

The transmitter reference design project includes code design, IP, and constraints, as shown in [Figure 3-1](#).

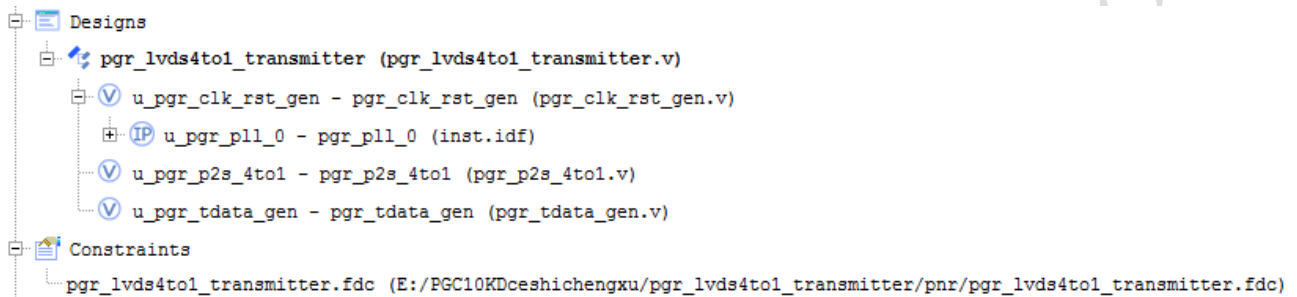


Figure 3-1 Reference Design Project Directory

➤ Code Design:

pgr_lvds4to1_transmitter: The top-level module of the transmitter, including external interfaces and module mapping.

pgr_clk_rst_gen: Clock reset module, generates transmitter clock and reset signals, produces a reference clock for receiver reset timing.

pgr_p2s_4to1: Implements parallel-to-serial data conversion and converts single-end to differential output.

pgr_tdata_gen: A module that generates four different types of data.

➤ IP:

This project requires the use of PLL IP, with specific configurations as shown in [Figure 3-2](#).

Figure 3-2 PLL Parameter Configuration

➤ Constraints:

`pgr_lvds4to1_transmitter.fdc`: Used for clock and pin constraints, different rates require corresponding clock constraints.

3.1.2 Receiver Reference Design Project Directory

The receiver reference design project includes code design, IP, and constraints, as shown in Figure 3-3.

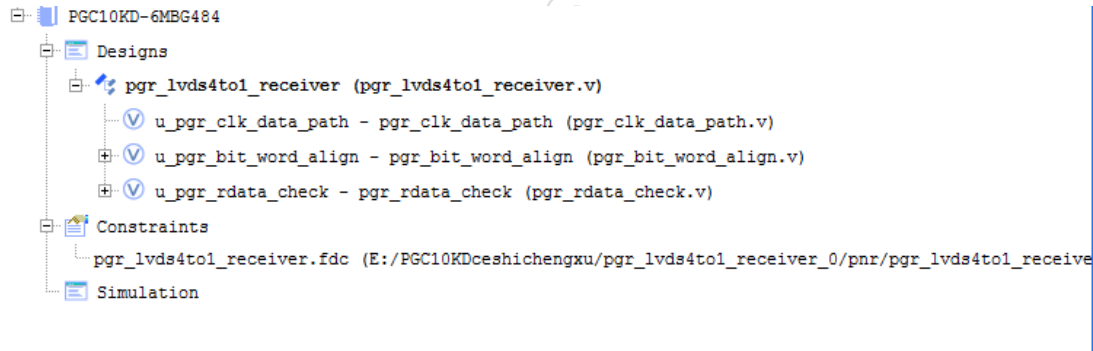


Figure 3-3 Reference Design Project Directory

➤ Code Design:

`pgr_lvds4to1_receiver`: The top-level module of the receiver, including external interfaces and module mapping.

`pgr_clk_data_path`: Clock reset module, generates receiver clock and reset signals, receives data and performs serial-to-parallel conversion.

`pgr_bit_word_align`: Soft logic can automatically complete word boundary and phase alignment.

`pgr_rdata_check`: Verification of the received data.

➤ Constraints:

pgr_lvds4to1_receiver.fdc: Used for clock and pin constraints, different rates require corresponding clock constraints.

3.2 Reference Design Board Validation

This reference design can be applied to single board self-loop verification as well as inter-board docking verification. Here, for inter-board docking verification, only part of the hardware functions on the test board are used, with the hardware environment block diagram used for verification shown in [Figure 3-4](#).

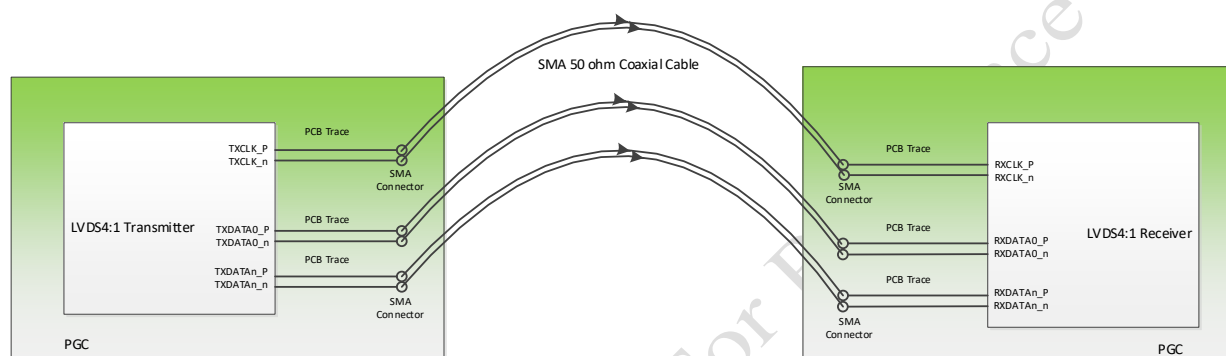


Figure 3-4 Verification Hardware Environment Block Diagram

The transmitter and receiver are connected by an SMA coaxial cable, with the IO Bank voltage for the LVDS transmitter/receiver set to 2.5V. Additionally, if the receiver uses external termination matching resistors, the on-die matching resistors must be turned off.

Disclaimer

Copyright Notice

This document is copyrighted by Shenzhen Pango Microsystems Co., Ltd., and all rights are reserved. Without prior written approval, no company or individual may disclose, reproduce, or otherwise make available any part of this document to any third party. Non-compliance will result in the Company initiating legal proceedings.

Disclaimer

1. This document only provides information in stages and may be updated at any time based on the actual situation of the products without further notice. The Company assumes no legal responsibility for any direct or indirect losses caused by improper use of this document.
2. This document is provided "as is" without any warranties, including but not limited to warranties of merchantability, fitness for a particular purpose, non-infringement, or any other warranties mentioned in proposals, specifications, or samples. This document does not grant any explicit or implied intellectual property usage license, whether by estoppel or otherwise.
3. The Company reserves the right to modify any documents related to its series products at any time without prior notice.
4. The information contained in this document is intended to assist users in resolving application-related issues. While we strive for accuracy, we cannot guarantee that the document is entirely free from flaws. Should any functional abnormalities and performance degradation arise due to deviation from the prescribed procedures outlined herein, our company will neither be held liable nor concede that such issues stem from product deficiencies. The solutions presented in this document are just one of the feasible options and cannot cover all application scenarios. Consequently, if users encounter functional abnormalities or performance degradation despite adhering to the prescribed procedures outlined herein, we cannot assure that such issues are indicative of product deficiencies.