GOLA Manual

Introduction

The GOLA is used to receive serial data from the GOL chip[1)].
The GOLA is based on the HOLA S-Link[2)] Link Destination Card (LDC) hardware. The FPGA content is reprogrammed so that the HOLA S-Link LDC becomes a GOLA.
The GOLA can be plugged on a ROMB (Read Out Mother Board) as if it was a S-Link LDC. However, the GOLA is **NOT** an S-Link implementation!

General

The serial data from the GOL chip will be received over a fibre optic cable. Only one fibre is used to send the data from the GOL to the GOLA; there is no return channel. The GOL to GOLA is in a test phase so to be cost effective a Custom Off the Shelf optical transceiver is used on both ends[3)]. At the GOL side, the transmitter part is used, and the receiver is omitted. On the GOLA side the receiver is used and the transmitter is omitted.
The GOL chip will send data in Ethernet mode, 32 bits at a time. The GOL clock will be 25.0000 MHz. This will result in a 1.0 GBd speed over the optical link after the four bytes are 8b/10b encoded[4)].
The GOLA board receives the serial data in a TLK1501 Transceiver. This transceiver will operate on 50.0000 MHz since the data arrives in chunks of 16 bits instead of the 32 bits at the GOL side.
The GOL sends the lower data bytes D[15:0] first, then D[31:16]. The GOLA expects the lower and upper data bytes to arrive consecutive. This means that there will always arrive an even number of consecutive 16 bit words in the TLK1501 on the GOLA.
The GOL can send Carrier Extend or an Error Propagation sequence[5)] to the GOLA. Both codes take up 4 bytes at a time in 32-bit mode and will be received as two 16-bit words in the TLK1501 on the GOLA.

Initialisation

There is no return channel from the GOLA to the GOL so initialising both ends of the link is a task for the system manager. The process should look like this:

- Power-up/reset the GOL and wait for the ‘Ready’ signal to arrive. This means that the PLL inside the GOL chip is locked and the GOL chip is ready to send data.
- The GOL will send Idles over the optical link to the receiver.
- Power-up/reset the GOLA. The TLK1501 on the GOLA will acquire lock onto the incoming serial data stream.
- When the PLL of the TLK1501 is locked onto the data stream, the TLK1501 will get into synchronization after receiving 3 consecutive Idles. Note that an IDLE

incorporates a 'comma character'[6] that makes it possible for the TLK1501 to byte align the incoming serial bit stream.
- The link is now considered to be 'up' and data can be send.

Reset

The GOLA can be reset following the protocol that is defined for the S-Link. The ROMB asserts URESET_n. This causes the GOLA to assert LDOWN_n. After initialisation of the GOLA, LDOWN_n will be de-asserted. When this is detected, the ROMB can de-assert URESET_n.
Note that a de-asserted LDOWN_n is no guarantee that the TLK1501 is locked and synchronized to the incoming data! When the incoming data stream is absent or is not sending Idles, then the TLK1501 might not be locked and/or might not be in its synchronized state (see also 'Initialisation' above).

Flow Control

There is no return channel from the GOLA to the GOL so there is no way to stop the GOL from sending data. Therefore the UXOFF_n signal on the GOLA has no meaning. However, the data that is received by the TLK1501 is placed into a small FIFO (63 words deep) to decouple the serial link clock from the 40 MHz interface to the ROMB. This FIFO makes it possible to implement a form of flow control for a few words.
When the FIFO becomes full then this is considered as an error condition (LDERR_n asserted).

Error Checking and Handling

The GOLA can check the integrity of the received data words with a CRC-32[7] check. If a Carrier Extend sequence is received from the GOL then the next data word is regarded as the CRC-32 over all previous data words. When the CRC-32 check fails then LDERR_n will be asserted.

When the GOL sends an Error Propagation sequence to the GOLA then this will also cause the LDERR_n signal to be asserted.
Note that the data bus of the TLK1501 contains 0xFEFE when an error propagation code group is received. Therefore it is most likely that the CRC-32 check will fail for the received data. This causes the LDERR_n to be asserted again!

Signal Descriptions for the GOLA

Pin Symbol	Pin Name	I/O	Description
LD[31..0]	Link Data output lines	Output from GOLA	Data present on these lines may be latched on the low-to-high transition of LCLK when LWEN_n is asserted (low). Synchronous with LCLK.
UXOFF_n	User Transmit Off	Input to GOLA	The GOLA has no return channel to the GOL so this signal has no meaning. However UXOFF_n can be used to signals the GOLA to stop transmitting data for a few words. Asynchronous. After UXOFF_n is asserted up to 3 words can be transferred before the transmition is stopped. When the GOLA FIFO fills up, an error condition occurs (See LDERR_n).
URESET_n	User Reset line	Input to GOLA	When low initiates a reset cycle. Asynchronous.
UTDO_n	User Test Data Out	Input to GOLA	Not used.
UDW[1..0]	User Data Width lines	Input to GOLA	Not used. GOLA transfers are always 32 bits wide.
LCTRL_n	Link Control line	Output from GOLA	Not used. GOLA always de-asserts (high) LCTRL_n.
LWEN_n	Link Write Enable line	Output from GOLA	When low indicates that valid data will be transferred to the ROMB on the low-to-high transition of LCLK. Synchronous to LCLK.
LCLK	Link Clock	Output from GOLA	Data is transferred to the ROMB on each low-to-high transition of LCLK when LWEN_n is asserted (low). This is a free-running 40MHz clock if LDOWN_n is de-asserted (high).
LDERR_n	Link Data Error	Output from GOLA	When low indicates that a data transmission error has occurred. There are three possible error sources: - Data Error - Received Error Propagation - CRC-32 Failed
URL[3..0]	User Return Lines	Input to GOLA	Not used
LDOWN_n	Link Down	Output from GOLA	When low indicates that the GOLA is not operational. Asynchronous. Is asserted (low) when the GOLA is undergoing a reset cycle. LDOWN_n will be de-asserted (high) when reset cycle is complete. The ROMB *shall* pull LDOWN_n low if the LDC is not present or not powered up.

Notes:

1) P. Moreira, T. Toifl, A. Kluge, G. Cervelli, A. Marchioro, J. Christiansen: GOL Reference Manual
2) Owen Boyle, Robert McLaren & Erik van der Bij: The S-Link Interface Specification
3) The GOLA will use an Infineon V23818-K305-B57 module. This is a Fibre Channel, 850 nm 1.0625 GBd transceiver with LC connector.
   The GOL will use a non-pluggable version of the same transceiver module: V23818-K305-L57.
4) The basics of the 8/10 bit coding are described in chapter 36 of the IEEE std. 802.3
5) The GOL in 32-bit Ethernet mode sends 4 code-groups for a Carrier Extend:
   <K23.7><K23.7><K23.7><K23.7> or 4 code-groups for a Error Propagation:
   <K30.7><K03.7><K30.7><K30.7>
6) IEEE std. 802.3, Chapter 36.2.4.4 handles the definition of the 'comma character'.
7) CRC-32 polynomial
   x^32+x^26+x^23+x^22+x^16+x^12+x^11+x^10+x^8+x^7+x^5+x^4+x^2+x^1+1