GLADIC - Latin America Group Development Integrated Circuit Open Source

Authors: Felipe Fernandes da Costa

Contents

Revision	Date	Author	Revised	Comments
0.1	12/11/2016	Felipe F. da Costa	-	Creating a file description IP

List of Tables

1 Introduction

1.1 Description(SpaceWire European Space Agency)

One of the principal aims of SpaceWire is the support of equipment compatibility and reuse at both the component and subsystem levels. In principle a data-handling system developed for an optical instrument, for example, can be used for a radar instrument by unplugging the optical sensor and plugging in the radar one.

Processing units, mass-memory units and down-link telemetry systems developed for one mission can be readily used on another mission, reducing the cost of development, improving reliability and most importantly increasing the amount of scientific work that can be achieved within a limited budget.

Integration and test of complex on-board systems is also supported by SpaceWire with ground support equipment plugging directly into the on-board data-handling system. Monitoring and testing can be carried out with a seamless interface into the on-board system.

SpaceWire is the result of the efforts of many individuals within the European Space Agency, European Space Industry and Academia.
The purpose of this Standard is:

1. To facilitate the construction of high-performance on-board data-handling systems
2. To help reduce system integration costs, and make it more modular
3. To promote compatibility between data-handling equipment and subsystems
4. To encourage reuse of data-handling equipment across several different missions
5. Support high frequencies work both receive and transmission

Figure 1 and Table 1 show the basic information and provide the basic view of top block. Since the propose is make a reduced and compact SpaceWire is decided to cut a few features from the it or delegate to another blocks making a smart use and provide another view point.

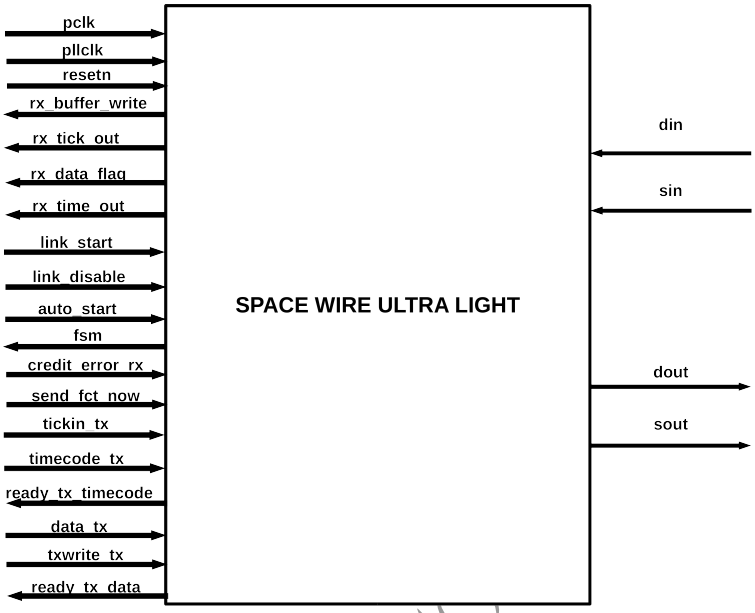

Figure 1: SpaceWireLight top block

Table 1: Pinout description

Signal name	Direction	Size	Description pin
pclk	input	1	Clock source used on fsm only.
pllclk	input	1	Phase Locked Loops.
resetn	input	1	Global negated reset.
rx_buffer_writer	output	1	write data on receiver fifo.
rx_tick_out	output	1	write data on receiver timecode.
rx_data_flag	output	9	Data captured from receiver.
rx_time_out	output	8	Timecode captured from receiver.
link_start	input	1	initialize codecs sequence start.
auto_start	input	1	initialize codecs sequence waiting.
link_disable	input	1	fsm to block codecs sequence start.
fsm	output	6	Provide fsm stats.
credit_error_rx	input	1	Signal is a overflow fifo rx.
send_fct_now	input	1	Signal is a read from fifo rx.
tickin_tx	input	1	Transceiver to start send a timecode.
timecode_tx	input	8	Timecode data to be delivered.
ready_tx_timecode	output	1	Let writer know timecode already send.
data_tx	input	9	Write data from fifo transceiver.
txwrite_tx	input	1	Transceiver to start send a data.
ready_tx_data	output	1	Writer know data already send.
din	input	1	Data bit encoded.
sin	input	1	Strobe bit encoded.
sout	output	1	Strobe bit encoded.
dout	output	1	Data bit encoded.

2 IEEE 1355-1995

2.0.1 Overview

For work operation of SpaceWire is used a IEEE pattern to provide a end to end connection both SpaceWires and ensure both are working and sending data to each other. These characters determine the control of real data across the codec and avoid errors like parity or overwrite information.

The following list gives a summary of the advantages of the IEEE 1355 technology:

1. Credit-based flow control on a per-link basis: this prevents packets from being lost in the switching fabric, which simplifies the higher layer protocols, since the retransmission of packets is not necessary, unless an error occurs.

2. Small protocol overhead: this makes the links very efficient, even for short packets.

3. Flexible packet format: this allows IEEE 1355 networks to be used as a carrier for other higher level protocols.

4. IEEE 1355 provides a set of lightweight protocols for bidirectional flow-controlled, point-to-point communication.

5. Low implementation complexity of IEEE 1355 interfaces: this enables the implementation of packet switches with a large number of ports.

6. Low latency and minimal buffering: the fast link level flow control of IEEE 1355 links enables the use of "wormhole" routing, which provides low switching latency and also requires minimal buffering in the switches.

2.0.2 Signal layer

DS-Links consist of four wires, two in each direction, one carrying data and one carrying a strobe, hence the term DS-Links (Data/Strobe). The data signal carries the serial bit stream, while the strobe signal changes state every time data does not change. This ensures that there is a transition on either data or strobe at the boundary of every bit frame. The data/strobe wire pair thereby carries an encoded clock, which can be simply recovered by generating the logical exclusive-or of the two signals.

The advantage of the two wire transmission of the DS-Link over the more traditional approach for a serial communication link using only one wire is the simple clock extraction.The Data/Strobe transmission scheme is less sensitive to signal skew than a system which simply transmits the serial data and the clock signal on separate wires. The DS-encoding provides a full bit period of skew tolerance. Due to the encoded clock, DS-Links can also auto-baud, the transmit rate can be varied as long as it does not exceed the maximum speed of the receiver.

2.0.3 Character Layer

The first bit of a character is the parity bit, followed by a control flag, which is used to distinguish between control and data characters. If the control bit is zero then it is followed by 8 bits of data, with the least significant bit being transmitted first. Control characters are 4 bits long and consist of a parity bit, the control/data bit which is set to 1 to indicate a control character, and 2 bits to indicate the type of control character. The Figure 2 hows the encoding of the DS-Link characters.

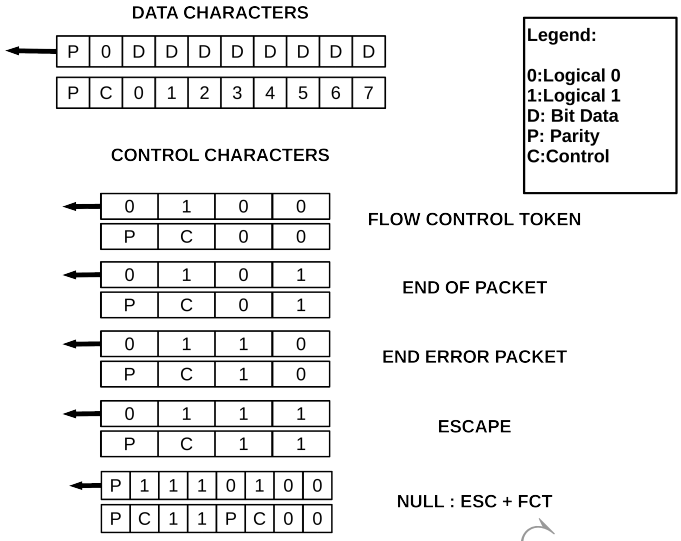

Figure 2: Character layer view

The parity bit covers the data or control bits in the previous character and the control flag in the current character. This allows the detection of single bit errors. Odd parity checking is used, where parity bit is set such that the bits covered, including of the parity bit itself, always contain an odd number of ones.

The normal character EOP is used to indicate the end of a packet. The character EEP character can be used to signal to indicate that an error has occurred. NULL characters are transmitted in the absence of other characters. This enables the detection of link failures, due to a physical disconnection. The NULL character also allows the parity of the EOP marker to be checked immediately.

2.0.4 Flow Control Token

IEEE 1355 links use a credit based flow-control scheme, which is local to the link and works on a buffer-to-buffer basis. The flow control mechanism ensures that no characters can be lost due to buffer overflow, which simplifies the higher levels of the protocol, since it removes the need for retransmission unless errors occur. From a system view, a DS-Link connection therefore appears as a pair of fully handshaken FIFO buffers, one in each direction. The smallest unit on which flow control is performed is called a flow-control unit, or flit. Each receiving link input contains a buffering for at least one flit. Whenever the link input has sufficient buffering available for a further flit of characters, a flow control character is trans- mitted on the associated link output. This FCT gives the sender permission to transmit a further flit. The transmitter maintains a flow control credit value, which is decremented when a data or a

packet terminator character is sent, and incremented by the flit size when a flow control token is received.

The flit size for the DS-Link is 8 characters. Therefore the receiver must have a buffer for at least 8 characters. However, because of the latencies inherent in DS-Link implementation and in order to sustain the full data rate over longer distances, a larger buffer is required, so that the character level flow control does not restrict the maximum bandwidth of the link. The requirement for continuous transmission is that the next FCT is received before the previous flit of eight characters has been fully transmitted, so that the link output is never stalled.

2.0.5 Link Start-up

After power-on, both ends of a link maintain their data and strobe outputs at low. When a link is started, it transmits NULL characters until it has received a character from the remote end. The link then sends a number of FCT characters, corresponding to the number of flits that fit in the receive buffer. The link then enters normal operation and can send data characters when flow control credit is available. This sequence ensures that the initial flow control characters are not lost, because the remote end is still being reset.

Some of the available DS-Link devices however send the flow control characters immediately when started, without waiting to receive NULL characters from the remote end. To avoid loss of FCT characters, both ends of the link have to be started up in the correct sequence under external control. If one end of a link is reset during normal operation, that end stops transmitting characters. The receiver on the other end of the link detects this as a disconnection error and also stops transmitting and resets itself. After a delay both ends of the link are ready to start normal operation again. This scheme effectively allows the two ends of the link to operate in different reset domains.

2.0.6 Error Detection and Handling

The DS-Link protocol allows the most common errors to be detected. The parity check will detect all single bit errors at the DS-Link character level. The physical disconnection of a link can also be detected, since each link output transmits a continuous stream of characters once it has been started.

The DS-Link characters contain a parity bit which allows single bit errors to be detected. Odd parity checking is used. The parity bit in a character covers the parity of the data/control bit in the same character, and the data or control bits in the previous character, as shown in Figure 3 below. This scheme allows any single bit error in any single bit of a character, including the control/data bit, to be detected even though the characters are not all the same length.

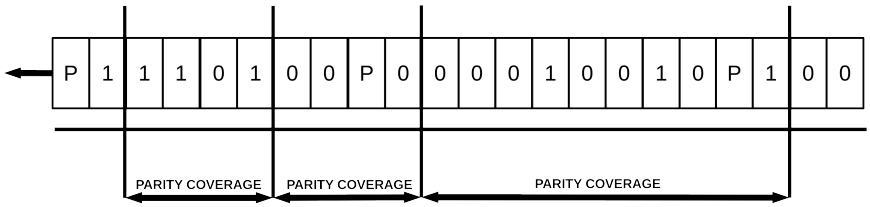

Figure 3: Parity Checker Example

3 Sub-blocks

3.0.1 Finite State Machine Control

The first block presented is FSM SpaceWire Light. This block is responsible to handle both receiver and transmitter.it provide a time machine state to give a flow control based in time to establish a connection between another SpaceWire. The Figure 4 provide a view from reduced signal propose to fsm block.

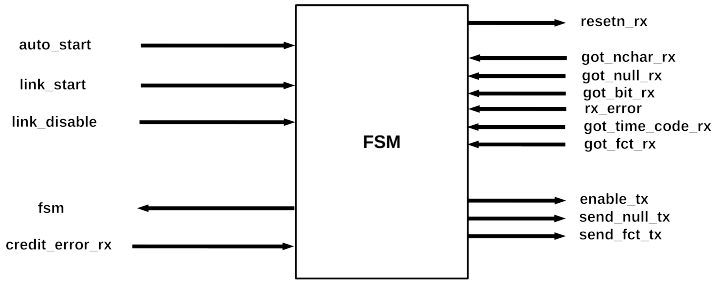

Figure 4: Finite State Machie SpaceWire

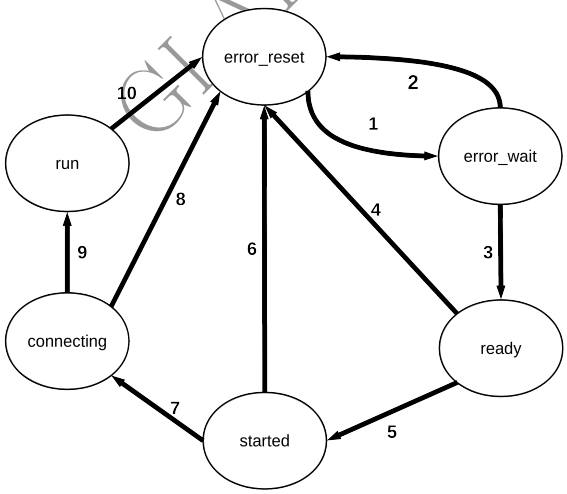

Figure 5: Finite State Machie SpaceWire

Figure 5 show detalied work state machine, where the definiton states a described below:

- error_reset
  - After a delay of 6400 ns (nominal) the state machine shall move to the error_wait state(1).
  - The error_reset state shall be entered after a system reset, after link operation is terminated for any reason or if there is an error during link initialization(2,4,6,8,10).
  - Transmitter and Receiver shall all be reseted in any time or state (1,2,3,4,5,6,7,8,9,10).
- error_wait
  - In the error_wait state the receiver shall be enabled and the transmitter shall be reset and allows the receiver to start the disconnection detection mechanism (after registering a transition on the D or S line) and to begin looking for the arrival of a NULL(got_null_rx).
  - If a NULL is received then the got_null_rx condition shall be set.
  - After a delay of 12800 ns (nominal) and the state machine shall move to the ready state(3).
  - If, while in the error_wait state, a disconnection error is detected, or if after the got_null_rx condition is set, a parity error or ESC error occurs, or any character other than a NULL is received, then the state machine shall move back to the error_reset state(2):(got_nchar_rx , rx_error , got_time_code_rx,got_fct_rx)
- ready
  - In the ready state the link interface is ready to initialize as soon as it is allowed to do so. The receiver shall be enabled and the transmitter shall be reset.
  - If a NULL is received then the got_null_rx condition shall be set.
  - The state machine shall wait in the Ready state until the the follow flags and then it shall move on into the Started state:
    - link_disable is a flag seted by software or hardware to indicate that the link is disabled. diagram(4).
    - link_start is a flag seted by software or hardware to start a link, where it cause the transition from the ready state to the started state(5).
    - auto_start is a flag seted by software or hardware to request the link to start automatically on receipt of a NULL(5).
  - In the ready state, a disconnection error is detected, or if after the got_null_rx condition is set, a parity error or escape error occurs, or any character other than a NULL is received, then the state machine shall move to the error_reset state (6):(got_nchar_rx , rx_error , got_time_code_rx,got_fct_rx).

- started
  - The started state enter from the ready state when the link interface is enabled.
  - When the started state is entered a 12800 ns (nominal) timeout timer shall be started.
  - In the started state the receiver shall be enabled and the transmitter shall send NULLs.
  - If a NULL is received then the got_null_rx condition shall be set.
  - The state machine shall move to the connecting state if the got_null_rx condition is set.
  - In the started state the sending from the transmitter of at least one NULL shall be requested before moving to the connecting state.
  - If, while in the started state, a disconnection error is detected, or if after the got_null_rx condition is set, a parity error or escape error occurs, or any character other than a NULL is received, then the state machine shall move to the error_reset state.
- connecting
  - The Connecting state shall be entered from the started state after a NULL is received (got_null_rx).
  - On entering the connecting state a 12800 ns timeout timer shall be started.
  - In the connecting state enabled and the transmitter is enabled to send FCTs and NULLs.
  - If an FCT is received (got_fct_rx condition true) the state machine move to the run state.
  - If, while in the connecting state, a disconnect error, parity error or escape error is detected, or if any character other than NULL or FCT is received, then the state machine shall move to the error_reset state (8):(got_nchar_rx , rx_error , got_time_code_rx).
- run
  - The run state enter from the connecting state; the Transmitter is enabled to send Time-Codes, FCTs, N-Chars and NULLs.
  - If the link interface is disabled, or if a disconnect error, parity error, escape error or credit error is detected , while in the run state, then the state machine to the error_reset state(10).

Inside FSM there are timers like decribed on states above where it give some time to receive and enable both receiver and transmitter 12800 ns and 6400 ns ; where these timers make part of normative. The diferent way adopted here

is to add 850 ns timer inside FSM instead leave it o receiver like described on normative. This timer is used on disconnection error and is linked with got_bit_rx.

The bit credit_error_rx is given by the overflow fifo receiver; this was made to simplify receiver block and avoid clock domain like disconection error timer. Signals resetn_rx and enable_tx have same function to reset both blocks and enable them in specific time. FSM signal provide to up interfaces information about state connecting only.

Signals send_null_tx and send_fct_tx are enabled according FSM is set to act and from what receiver get from DS-link.There are anothers signals described in normative but they removed cause in design understanding these signals is more valuable to stay inside of FSM and let transmitter handle it by his own state machine.

3.0.2 TX SpaceWire Ultra Light Core

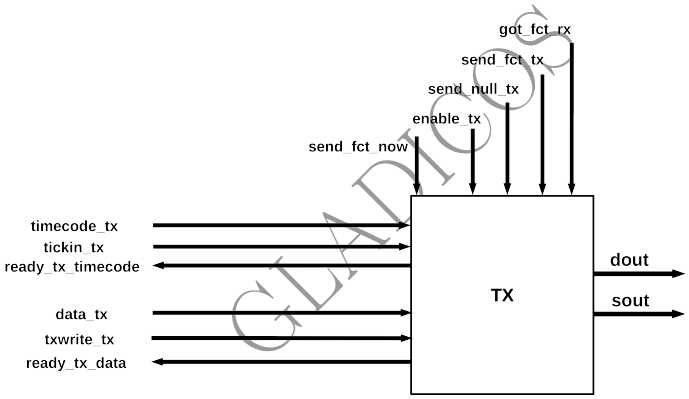

Figure 6: Transmitter

Transmitter is used to encode information "IEEE-1355" to be delivered accross the line.. It receives N-Chars for transmission from the host system. If there is neither a Time-Code, FCT nor an N-Char (data, EOP or EEP) to transmit, the transmitter sends NULL. The transmitter sends N-Chars only if the up layer system at the other end of the link has room in its host receive buffer. This is indicated by the link interface on another side sending an FCT, showing that it is ready to accept another 8 N-Chars. The transmitter is responsible for keeping track of the FCTs received and the number of N-Chars sent to avoid

input buffer overflow at the other end of the link. To do this the transmitter holds a credit count of the number of characters it has been given permission to send.

The transmitter can be in one of four possible states:

- tx_spw_start: Start to send NULLs only.
- tx_spw_null: It can only send NULL on the link. It does not read N-Chars from the up layer interface. It does not accept an order to send FCT from the level above. It does not send Time-Codes.
- tx_spw_null_fct : It sends FCT or NULL but still does not read N-Chars from the up layer interface. It does not send Time-Codes.
- tx_spw_full: Normal behaviour, sending NULLs, FCTs, Time-Codes and N-Chars.

Signals in Figure 6 have the follow description below:

1. enable_tx (negedge): The transmitter does nothing.
2. got_fct_rx (posedge): This signal is responsible to increment N-Chars counters slots open in another SpaceWire. Each FCT correspond in 8 open slots implies the max slots to till 56 positions. On Transmitter this value is seted to 7 FCTs after a disable block.
3. send_fct_tx (posedge): Signal is used to enable transmitter to send FCTs.
4. send_null_tx (posedge): This signal after transmitter is enabled according configuration is seted (link_enable / auto_start)
5. send_fct_now (posedge): Signal is enabled when receiver fifo is read 8 times.
6. tickin_tx(posedge): Signal used to enable transmitter send data previous seted from timecode_tx; At once Time-Code is sent the signal time_code_ready is seted(posedge).
7. txwrite_tx(posedge):Signal used to enable transmitter send data previous seted from data_tx; At once N-Char is sent the signal ready_data_tx is seted(posedge).

The transmitter can operate at any data signalling rate from the minimum to the maximum possible. The transmit clock is responsible for producing the variable data signalling clock signals used by the transmitter. The transmit clock signals are typically derived by dividing down the local system clock or a phase locked loop multiple of the local system clock.

3.0.3 RX SpaceWire Ultra Light Core

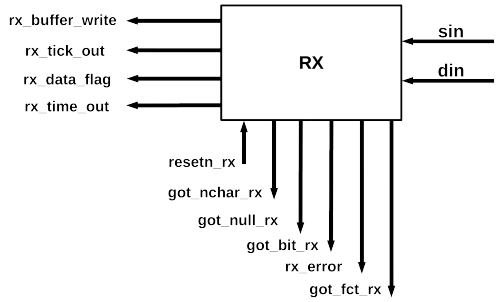

Figure 7: Receiver

The receiver is responsible for decoding the DS signals (Din and Sin) to produce a sequence of N-Chars (data, EOP, EEP) that are passed on to the host system. It also receives NULLs, FCTs and Time-Codes. NULLs represent an active link. They are flagged to the exchange-level state machine(FSM) but are ignored otherwise. When an FCT is received the receiver informs the transmitter so that it can update its credit count accordingly. All other control characters received are flagged to the state machine. The receiver ignores any N-Chars, L-Chars, parity errors or escape errors until the first NULL is received. The disconnection detection mechanism within the receiver is enabled as soon as the first bit arrives (first transition detected on D or S inputs to receiver).

This Block dont have state machine make it be more simple and use only clock recovery. This approach try to avoid the use inside the block clock system and try only to indentify character recovered from DS-Link.

Signals in Figure 7 have the follow description below:

1. resetn_rx (negedge) : Disable receiver interface according FSM timers.
2. got_nachar_rx (posedge) : Flag used to notice FSM got a NChar.
3. got_null_rx (posedge): Flag used to notice FSM got a NULL.
4. got_bit_rx (posedge) : Flag used to notice FSM receiver is active.

5. rx_error (posedge) : Flag used to detect parity error, wrong character combination.
6. got_fct_rx(posedge) : Flag used to notice FSM/Transmitter FCT got.
7. rx_buffer_write(posedge) : Make enable fifo to write data from rx_data_flag.
8. rx_tick_out(posedge): Make enable fifo to write data from rx_time_out.

4 SpaceWire Ultra Light

SpaceWire Light must be expected the follow characteristics:

- Receiver/Transmmiter support High frequencies on 300 Mbps.
- Low Density on blocks developed.
- Reduced functions avaiable on ESA "European Space Agency" standart.
- Provides another view point to improve specification.

5 References

SpaceWire – Links, nodes, routers and networks ; ECSS-E-ST-50-12C 31 July 2008
IEEE Standard for Heterogeneous InterConnect (HIC) (Low-Cost, Low-Latency Scalable Serial Interconnect for Parallel System Construction) ; IEEE Std 1355-1995 (Adopted by ISO/IEC and redesignated as ISO/IEC 14575:2000)