SIMPLEX MODE IN SPW TECHNOLOGY

Session: SpaceWire networks and protocols

Short Paper

Eugenue Yablokov

Saint-Petersburg University of Aerospace Instrumentation. 67, B. Morskaya, Saint-Petersburg, Russia

E-mail: kabalenator@gmail.com

ABSTRACT

A very important task in developing SpaceWire interconnections is the development of the simplex mode. Reducing number of lines is a good solution for working with devices, which are not designed for working in full duplex. The simplex mode also reducing the number of lines, thus reducing square and weight, which is very useful on board of spacecraft. For example, simplex mode can be used working with video camera, the block will only receive information from camera. Simplex mode can be also used for control the block, for example moving source of light etc. All data will be transmitted using one Tx/Rx pair instead of two, thus decreasing the cost of the cable.

The SpaceWire controller can work in two possible directions of the simplex mode – transmitting and receiving. Transmitting part sends symbols due to standards and from time to time use the special mode of reconnection. The receiving part establishes connection and detects errors. The FCT symbols are not sending, so the receiving part is always ready to receive the symbol of data.

Due to changing the number of lines, however, several parts of the standard SpaceWire were changed. Sending FCT symbols for reserving eight words of buffer were not possible, so the credit system was not used. The problem of connection is very sufficient, transmitting part doesn't know if connection is established. If an error occurred during transmission, the receiver part was sent to reset state and after reset it had to establish connection again, whereas the connection had to be established on speed 10 MHz. These problems are solved in our SpaceWire controller.

Simplex mode is designed for one-side transmission or receiving data. This mode let us minimize the number of transmitting or receiving cable, reduce the number of gates of the block, thus reducing the weight of the whole block. The block with simplex mode included, give us the number of advantages, though there are some problems to solve, such as establishing the connection, because in normal block connection is established using two directions. The crediting system can't be used as it is in SpaceWire specification.

Our modification allows to use the simplex mode. Considerable changes in the standart SpaceWire were not made. Just one part of the standart Spacewire is modified – state machine. The new state machine will help us to use the simplex mode depending on signals designed for simplex mode. Using only two new signals – tx_simplex_enabled and rx_simplex_enabled will allow us to turn this mode on. If this two signals will not be set to active level – state machine will work as an ordinary state machine of the SpaceWire standart. On figure 1 our new state machine is listed.

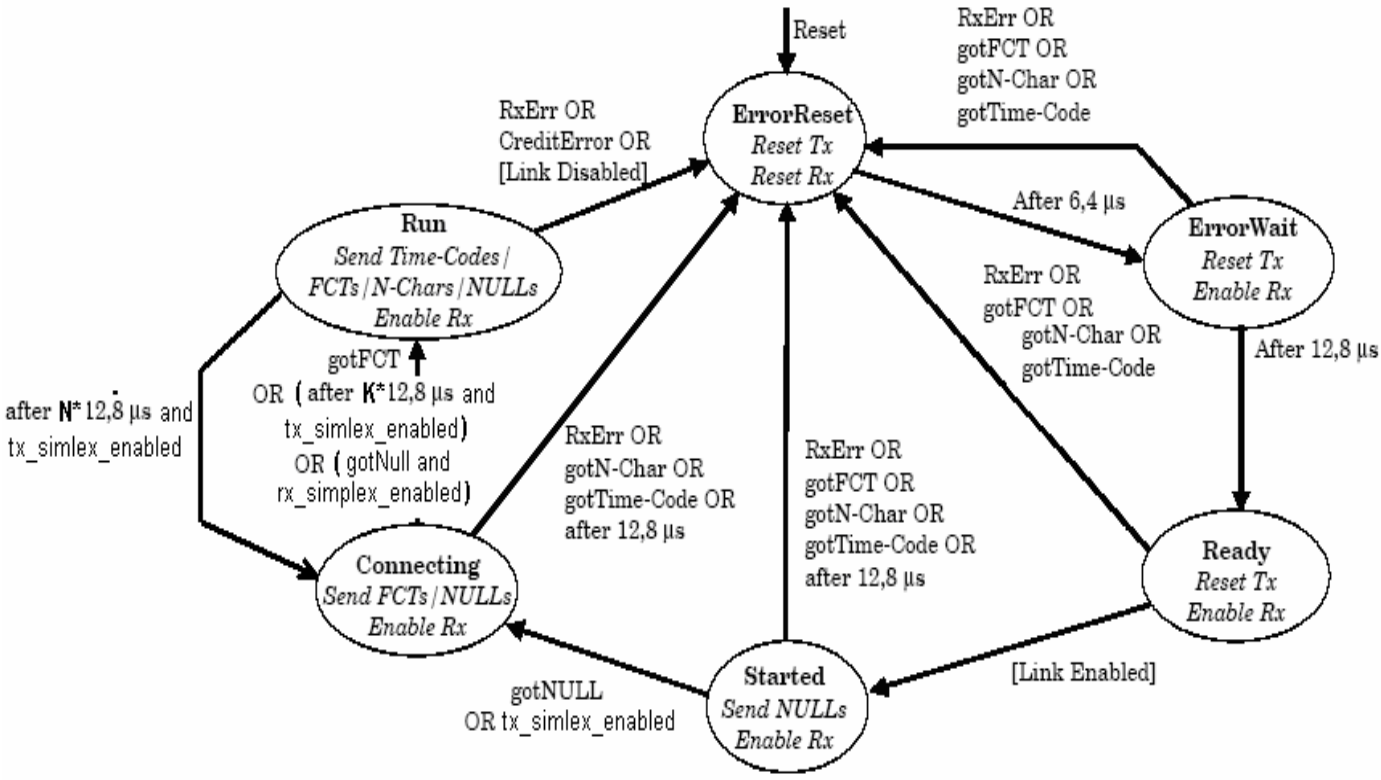

Figure 1. Modified state machine

All modifications of this state machine are made after state "Started". Also signal Link Enabled will be formed a little bit different from original state machine.

As we mentioned before – there are two types of the simplex mode – transmitting simplex mode and receiving simplex mode. Let us examine the behavior of the state machine in two types of simplex mode.

Receiving simplex mode is enabled by setting the active level of signal on line rx_simplex_enabled. In this mode the block will only receive data and/or time codes. Firstly state Machine will be in ErrorReset State. After 6.4 microseconds the block will go to ErrorWait state. After 12,8 microseconds it will be in Ready state. Link Enabled simplex in the receiving simplex mode is set only if the Null symbol is received and the AutoStart signal is in active level. The LinkStart signal is not used in receiving simplex mode. Thus, after the receiving of the NULL symbol the state machine will go to the state Started and then to Connecting. There is no sense waiting the FCT symbol in this state, because no data will be sent. So, after the block goes to Connecting state it goes to the Run state. In this state the block will be until the signal Link Disabled is set, or any other error occurred, like in the original state machine. In conclusion I can

say that state machine of the receiving simplex mode is simple – after the Null symbol is received and Autostart is set – it goes directly to Run state.
Transmitting simplex mode is enabled by setting the active level of signal on line tx_simlex_enabled. Firstly state Machine will be in ErrorReset State. No errors can be occurred in this mode. The only way to go to ErrorReset state is setting the Reset signal or Link Disable signal in Run State. After 6.4 microseconds the block will go to ErrorWait state. After 12,8 microseconds it will be in Ready state. The block will be in this state until the signal Link Start is set to active level. In the transmitting simplex mode AutoStart signal is not used. After the signal Link Enabled is set to active level – state machine go to Started state. The state machine goes to Connecting state at once, because there is no receiving channel. State machine will stay in this state for a sufficient time (K*12,8 microseconds). This time can be set by a designer, the only condition is that a NULL symbol must be sent on frequency 10 MHz. Such time-consuming state is made because the receiving must detect the first NULL. This state also will be used for reconnecting, which will be described further. After K*12,8 microseconds the state machine goes to Run state. In this sate the transmitter can send data and time codes. FCT codes are not sent. In this state the block stays for N*12,8 microseconds. After this time state machine goes to Connecting State and the transmitter begin to send only NULL symbols on the frequency 10 MHz. This period is called the period of reconnecting and is made for the receiving block. If an error occurred in the receiving block – the receiving block will go to the ErrorReset state. After some time it will go to the Ready State and will start the Connection only if the transmitting block will send the NULL symbol. The transmitting part doesn't know the situation in the receiving part. That's why such periods of reconnection are made. If an error occurred, the connection can be established again. Maximum time of reconnection period is 12,8*N+12,8*K microseconds.
In conclusion we can say that if the simplex mode enabled the state machine of the receiving part can be in the Run state till doomsday (if no errors occurred), whereas the state machine of the transmitting part will be changing its state, moving from Connecting to Run and from Run to Connection.
The FCT signal is not used at all, that's why on the receiving part the big-sized buffer have to be used, or the reading speed from the buffer must be higher than the writing speed from the receiving part.
Siplex mode can be used as the mode of the block, or the can be the block, using just one channel, receiving or transmitting. If the block will use only one channel, the size of block will be reduced significantly, because the other channel will not be even sinthesized. The use of simplex mode will reduce the power consumption of the block, also because the second channel will not be working.
The maximum traffic capacity of the SpaceWire channel is 320 Mbit per second on the frequency 400 Mhz. This traffic capacity can be achieved in half-duplex mode (when one channel is sending data symbols only and the other – FCT, Time and Null symbols). In full duplex mode traffic capacity is reduced to 305 because of the FCT symbols in each channel. Because of the period of reconnection the traffic capacity is lower in simplex mode than in half duplex mode. On Figure 2 the traffic capacity of the channel in simplex mode (we are using N=8 and K=1).

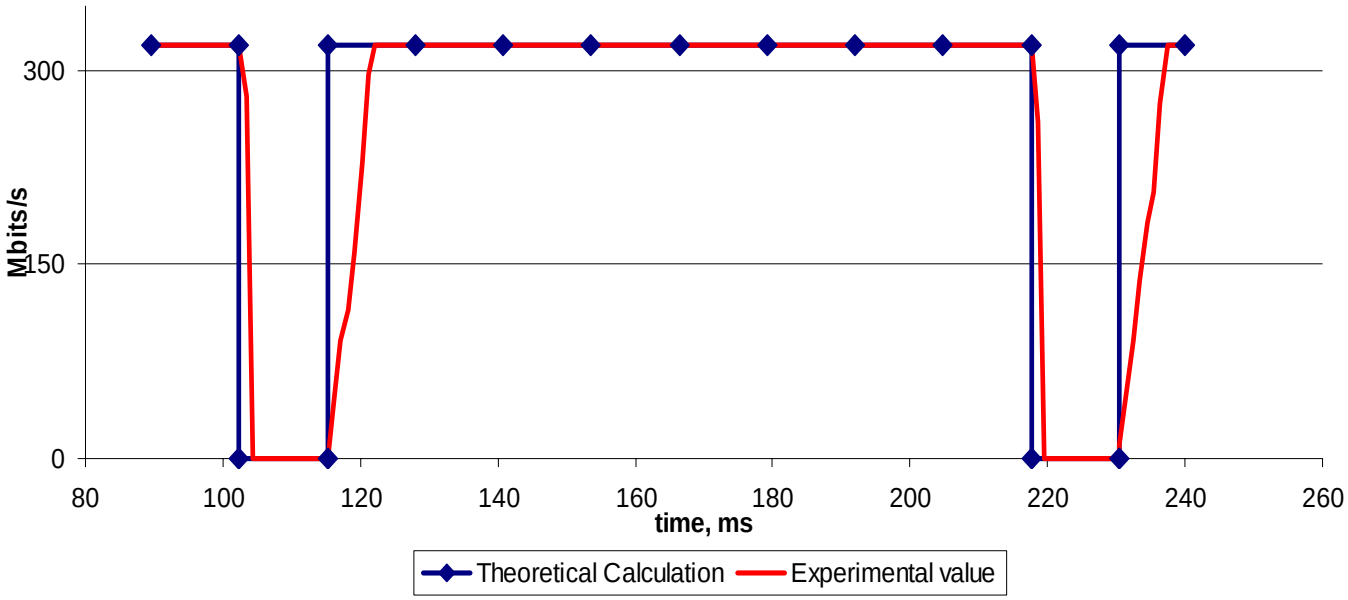

Figure 3 Traffic capacity in simplex mode (frequency 400 MHz).

The block in state Run is sending Data for a period 102,4 microseconds, and is sending Null symbols for 12,8 microseconds. As we can see from practical value – the traffic capacity is not going instantly to 0, the transmitter is sending some data, left in transmitting buffer, and after this it falls to zero. But the frequency is not 10 MHz, the PLL also can't set 10 MHz in a moment, frequency goes down smoothly. When State machine of the transmitting part goes to state Run – traffic capacity also don't go to maximum because of the PLL. Let us look through average statistical value of different modes of transmitting Data. The traffic capacity in simplex mode is significantly lower then in other modes because of the period of reconnection (figure 3).

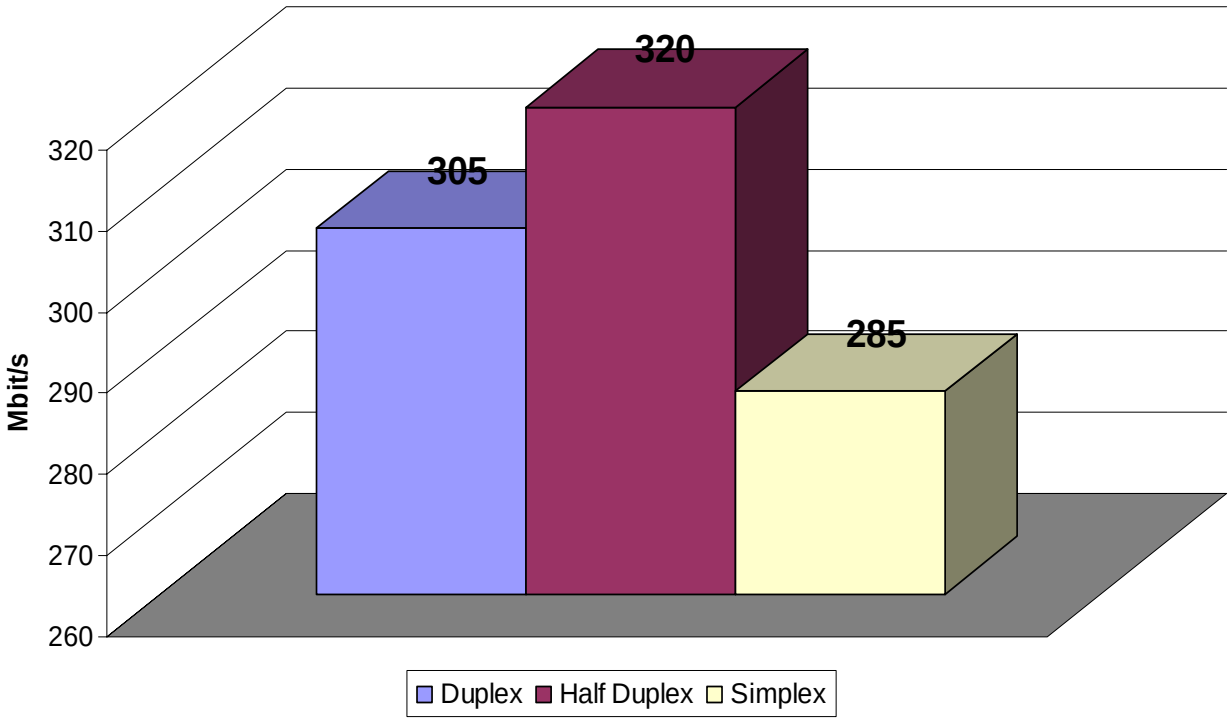

Figure 3. Traffic capacity of the block in different modes

The more is the period of sending data in simplex node and the less is the period of the reconnection – the higher is the traffic capacity, but that will increase the number of data lost if an error occurs. Because of this – parameters N and K must be set due to situation.