1.3 Acronyms and abbreviations

Acronym/Abbreviation	*Description*
BER	Bit Error Rate
CCSDS	Consultative Committee for Space Data Systems
CODEC	COder-DECoder.
DDR	Double Data Rate
ECSS	European Collaboration for Space Standardization
EEP	Error End of Packet
EMC	Electromagnetic Compatibility
EoP	End of Packet
ESA	European Space Agency
FCT	Flow Control Token
FIFO	First In First Out
IEEE	Institute of Electrical and Electronics Engineers
JAXA	Japan Aerospace Exploration Agency
LVDS	Low Voltage Differential Signaling
NASA	National Aeronautics and Space Administration
NCHAR	Normal Character
NEC	Nippon Electric Company
PCB	Printed Circuit Board
PnP	Plug and Play
PVS	Protocol Validation System
RKA	Russian Federal Space Agency
SpW	SpaceWire
SUAI	St Petersburg State University of Aerospace Instrumentation
TBC	To Be Confirmed
TBD	To Be Defined
VHDL	VHSIC Hardware Description Language
WG	Working Group

2 User Requirements and Analysis for Simplex and Half-duplex SpaceWire

This section gives a brief introduction to the solution proposed for Half Duplex by 4Links and for Simplex by SUAI and contains the requirements for Simplex and Half Duplex SpW which 4Link collected by the Space Industry.

2.1 Introduction to Low mass SpW

SpaceWire provides full-duplex operation on four pairs of wires, allowing for equal data rates in each direction. In addition to data, the 'reverse' direction also provides flow control and control of link recovery after a transmission error. Some applications, however, do not need equal transmission rates in both directions –data from sensors or to actuators, for example. In this case the full-bandwidth reverse channel is seen as unnecessary and the value of the mass of the two pairs of wire for the reverse direction is open to question.

Currently two proposed SpW Evolutions work towards mass reduction for this type of applications:

- Simplex SpW: Proposed by SUAI. It specifies transmission in one direction only. There are no Flow Control Tokens ([RD, 4]) and therefore the flow of data cannot be controlled which may result in data losses.
- Half Duplex SpW: Proposed by 4Links. It specifies Half-Duplex bidirectional communication in which the two ends of the SpW link share a common D-S link on which they transmit alternatively ([RD, 2]). Half Duplex supports Flow Control and offers all of the Full Duplex SpW features.

2.1.1 Half Duplex SpW

An alternative approach is to use just two pairs of wires for both directions – half-duplex operation. The wires must be shared between both ends of the link and a mechanism for synchronizing transmission/reception introduced. The benefit of such an implementation is retention of flow control, error recovery and header configuration whilst the bandwidth may be almost entirely dedicated to data flow in one direction. As well as retaining the benefit of the reverse direction, all the other benefits of SpaceWire are retained, such as the modularity and simple mapping of other protocols (CCSDS, Ethernet, ...).

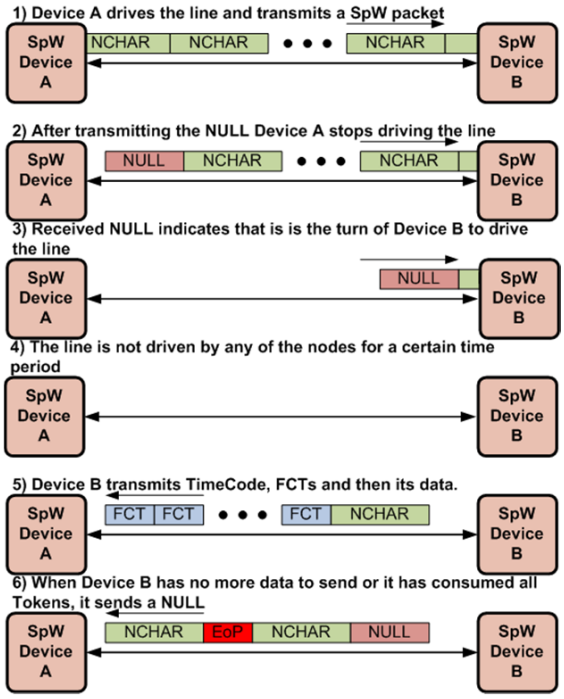

Figure 1: Half Duplex SpW functionality

Figure 1 presents an example in which two devices operate using Half Duplex SpW.

1) Initially, device A has received FCTs from device B and is transmitting a SpW packet.

2) When all FCTs have been consumed (or there are no more NCHARs/EoP) for transmission, device A sends a NULL indicating that it wishes to release the bus.
3) The received NULL is received by device B and signals that it is the turn of device B to drive the Half Duplex link.
4) For a certain amount of time the shared link is not driven by any of the devices in order to provide for relaxation time for the signals.
5) Device B starts driving the link and it sends FCTs (if it has consumed the received NCHARs) and then its NCHARs, if there are data to transmit. If there are not data to transmit, it sends a NULL and it is the turn of device A to drive the link again. In addition, if a Time Code is pending for transmission at device A, this is sent over the link before any other character i.e. before the FCTs.
6) After having sent the Time Code and FCTs device B sends its NCHARs/EoPs until the FCTs it has received are consumed. It then sends a NULL indicating that it is the turn of device A to drive the bus again.

SpaceWire is easy to implement and very well suited to the construction of highly fault tolerant networks. Normal, full-duplex, SpaceWire is ideal for bi-directional traffic but is under-utilized for (largely) unidirectional traffic such as that from sensors or to actuators. Nevertheless, a reverse data flow is useful for control and status even it is low volume.

Analysis of the Half-duplex SpaceWire shows that, for a given data throughput, it offers a significant mass reduction in asymmetric traffic flow situations. A further benefit is the reduction in the number of pins required on a chip for half-duplex operation.

A more radical possibility, not carried into the SpaceWire Evolutions project, would be to use two half-duplex links in place of a single full-duplex link. This would give a similar bidirectional data rate (and a considerably higher unidirectional data rate) for a 40% reduction in cable mass – and also provide redundancy for fault tolerance.

2.1.2 Simplex SpW

For low mas SpaceWire SUAI has proposed Simplex SpW in which the link is asymmetric, consisting of a transmitter and a receiver and information flows in one direction only. The transmitter can only send NCHARs, EoP/EEP and Time-Codes. It does not send FCT since the remote end cannot send any data. The receiver can only receive NCHARs, EoP/EEP and Time-Codes and does not send anything since there is no path from the receiver to the transmitter. This means that the SpW flow control mechanism is not used at all in Simplex mode. This implies changes in the SpW state machine and functional model since:

- FCTs are used in link initialization, so how can the devices know if the link has been initialized?
- The data transfer mechanism is based on flow control so it needs to be modified

To this respect SUAI proposes:

- Modification of the link initialization procedure
- Modification of the data transfer mechanism

Since in Simplex the communication is unidirectional only the receiver can synchronize to the transmitter. Furthermore, the absence of FCTs in the link means that another character shall be used for initialization. Therefore, initialization can be based on the reception of Time-Codes, NCHARs or NULLs.

- NCHAR based initialization: with this approach it is possible that a receiver may have disconnected and be in synchronization phase while the transmitter has already started transmitting a packet. This means that the first NCHAR received may be perceived as routing information (Figure 2) and if the receiving device has other SpW links the rest of the packet may be erroneously routed in the network and disrupt links which carry sensitive traffic.
- Time-Codes based initialization: Simplex is oriented for use in simple devices and is not foreseen to be used in Time-Code masters. However, in order to support this method requires modification at both the receiver and at the transmitter and adds unnecessary complexity.
- NULL based initialization: With this approach the receiver synchronizes upon the reception of a NULL. This approach does not cause problems to the network operation, requires only simple changes at the receiver and this is the solution proposed by SUAI.

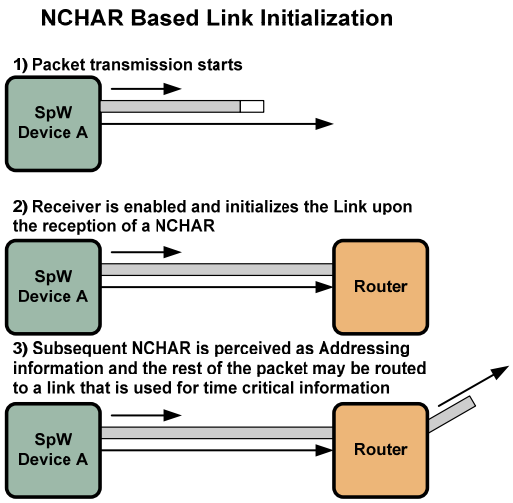

Figure 2: NCHAR based initialization

Since the transmitter cannot be aware of the receiver's state the scenario described in Figure 3 is theoretically possible. In this example the transmitter is transferring-back-to-back packets without any NULLS in between. Since receiver initialization is based on NULLs the receiver does not synchronize and transmitted information is lost

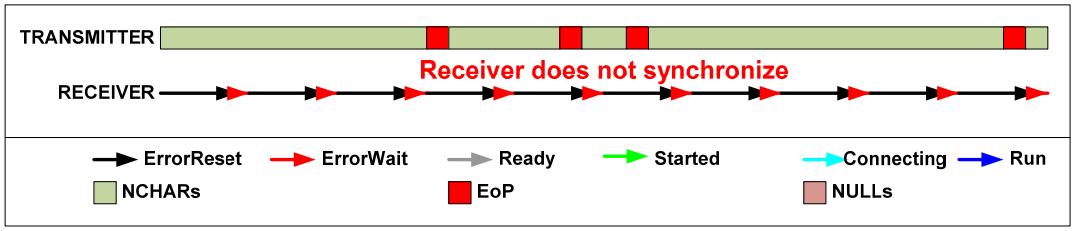

Figure 3: Problem in Simplex synchronization in case of back-to-back packets transmission

In order to overcome this problem, SUAI proposes a simple functional change in the Simplex transmitter's state machine which ensures that NULLs are inserted between packets. The transmitter time, is divided in 12,8 us multiples and for Nx12,8 us the transmitter can transmit data and for Kx12,8 us it transmits NULLs only, thus ensuring that NULLs are inserted in the traffic flow and the receiver can synchronize (shown in Figure 4).

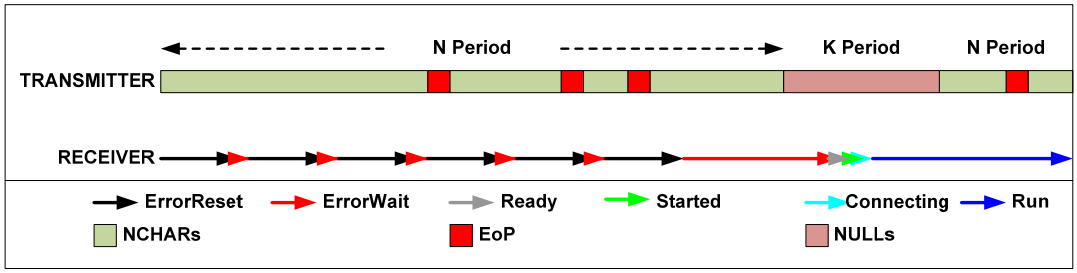

Figure 4: SUAI's proposed modification in order to ensure synchronization in Simplex mode

The SUAI proposal concludes with the state machine shown in Figure 5 which is capable of supporting both Full Duplex and Simplex modes and requires simple changes in the SW implementations.

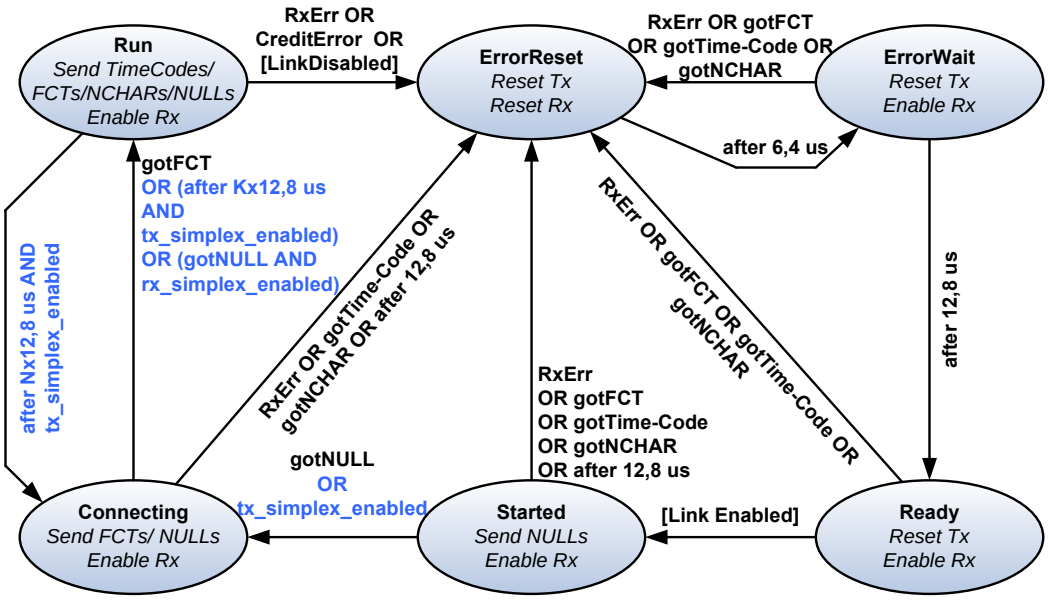

Figure 5: SUAI proposed state machine for Full Duplex and Simplex SpW

2.2 Requirements Collection Methodology

Initial requirements were summarized in the proposal. These were extracted and sent to the State University of Aerospace Instrumentation in St Petersburg (SUAI), who proposed the Simplex SpW, and to companies who use SpaceWire and have contributed to activities of the SpaceWire Working Group.

Response was not, unfortunately, received from SUAI. Responses were received, however, from Honeywell, NEC, RUAG and Thales Alenia Space.

Astrium were not included in the gathering of requirements phase because Astrium were to review the collected requirements, and this document includes additional requirements and comments from Astrium, based on the inputs from the other users.

The requirements resulted in the need for a number of additional Definitions, such as Interoperability, Reserved, and Options. Initial suggested definitions are included in this document, but these need to be discussed and considered carefully within the project, and so the important aspect concerning the definitions is that they are required for unambiguous understanding of the revised standard.

Some of the Requirements are fundamental, others are intended to satisfy these fundamental requirements and are essentially implementation requirements. In tabulating the Requirements, this distinction is made explicit.

2.3 Collected User Requirements and Analysis (4Links)

2.3.1 Fundamentals: Form of Standard, Interoperability and Definitions

These are repeated from the fundamental requirements given in D2

2.3.1.1 Form of SpaceWire 1.1 additions and changes to the SpaceWire standard

Source	**Astrium**
Requirement	A compliance test set should be defined for ECSS which allows to assess compliance of a device. The same for SpaceWire 1.1.
	Analysis
The current ECSS-E-ST-50-12C standard includes several instructions of how something should be done, but without compliance criteria that can be measured. Correcting these is outside the scope of the current project, but all changes and additions to the standard for SpaceWire 1.1 should be written such that compliance criteria can be measured.	

2.3.1.2 Interoperability

Source	**SUAI, summarized in the proposal**
Requirement	SpaceWire 1.1 implementations shall be interoperable with implementations that comply with ECSS-E ST-50-12C
	Analysis
(4Links) This requirement is universal across all the SpaceWire evolutions. It does depend, however on the definition of interoperability, and there is no such definition in ECSS-E-ST-50-12C.	

2.3.1.2.1 Offered Definition of Interoperability

Source	**4Links**
Offered Definition	A device built to SpaceWire 1.1 1. may generate traffic that is ignored by a device built to ECSS-E-ST-50-12C; 2. may generate (as part of an initialization or recovery procedure) traffic that causes a device built to ECSS-E-ST-50-12C to disconnect, but shall not, except in such an initialization or recovery procedure, generate such traffic. 3. shall not generate traffic that a device built to ECSS-E-ST-50-12C will interpret as valid traffic but with different semantics from SpaceWire 1.1.
	Analysis
(4Links) The offered definition is specific to SpaceWire Evolutions and would be better if made totally generic to SpaceWire standards. The wording may not be adequately precise and needs to be considered carefully. (Astrium) “Totally agree” with this reservation. Also comment: (Astrium) The compliance definition is not clear since the implementation will have some limits. (4Links) Agree entirely with the lack of clarity of the suggested definition. (Astrium) A compliance test set should be defined for ECSS which allows to assess compliance of a device. The same for SpaceWire 1.1. (4Links) This comment generates the next Requirement (4Links) If there is a general ECSS definition of Interoperability which covers differences 1. within the Normative clauses of a standard? 2. between Normative and Optional clauses of a standard? 3. between Normative and Reserved features of a standard? 4. between different versions of a standard ? or if there is a definition in other standards, elsewhere than ECSS, that we could use, it might be preferable to inventing our own	

2.3.1.2.2 Interoperability with existing SpaceWire implementations

Source	**Honeywell, TAS, RUAG**
Implementation Requirement	SpaceWire 1.1 implementations should, in special cases, be interoperable with existing implementations of ECSS-E-ST-50-12C Interoperability with existing implementations of ECSS-E-ST-50-12C will be aided if SpaceWire 1.1 routing switches can disable propagation of all Time Codes, and can disable propagation of those "Time Codes" whose control flags are non-zero, from each output port independently.
Analysis	
(4Links) This enables SpaceWire 1.1 networks to include legacy nodes that treat non-zero control flags for Time Codes as if they were zero. The requirement comes from the following requests: (Honeywell) Aeroflex-Gaisler have raised some backward compatibility issues about time code character propagation in legacy routers, but (as far as I can determine) those issues are related to specific implementations rather than the SpaceWire standard (i.e. the implementations are more restrictive than the standard in propagating time code characters). (TAS) Ok the mechanism is not simple but seems to work (in addition it seems backward compatible with the ATMEL AT7910E which can be configured to discard incoming time codes whose control flags are not 00 (4Links) In principle, interoperability should not depend on implementation choices of existing devices. The issue is ambiguity of the interpretation of "Reserved" in ECSS-E-ST-50-12C, not helped by there being no definition. As a result, incompatible implementations can be built which all have valid claims to comply with the standard, and yet which need special features of third-party implementations (such as described by TAS) to make them interoperable. 4Links suggests defining "Reserved" in SpaceWire 1.1, and suggests that, in this specific case only, the implementation described by TAS in the ATMEL AT7910E is adopted. These are included as this Requirement and Definitions of Reserved and Option (Astrium) Does it means that SpaceWire 1.1 functionalities stay usable in a ECSS-E-ST50-12C compliant network ? or they could be degraded ? (4Links) Legacy devices not designed for SpaceWire 1.1 cannot, in general, use SpaceWire 1.1 capabilities. What this requirement ensures is that devices that fail to decode the Time Code control bits fully are not corrupted by Distributed Interrupts nor by multiple Time Codes.	

2.3.1.2.3 Offered Definition of Reserved

Source	**4Links, derived from comments from Honeywell, TAS, RUAG**
Definition	"Reserved" (4Links) The following definition is offered To be compliant with this version of this standard, implementations: 1. Shall not generate codes that are defined as Reserved 2. Shall ignore and discard any received codes that are defined as Reserved (without signalling that an error has been received)
Analysis	
(4Links) As above, the issue raised is ambiguity of the interpretation of "Reserved" in ECSS-E-ST-50-12C, not helped by there being no definition. As a result, incompatible implementations can be built which all have valid claims to comply with the standard, and yet which need special features of third-party implementations (such as described by TAS) to make them interoperable.	

2.3.1.2.4 Placeholder Definition of Option

Source	**4Links, derived from comments from Honeywell, TAS, RUAG**
Definition	"Option" TBD
Analysis	
(4Links) This is inserted as a placeholder, because it has similarities with Reserved but Options apply to different implementations within a version of a standard whereas Reserved is intended for different versions of a standard.	

2.4 Requirements for Simplex

2.4.1 Simplex not required if half-duplex is possible

Source	Honeywell, RUAG, TAS, Astrium
Requirement	A reverse channel is required for control, which is provided by half-duplex and not by simplex, and so there appears to be a consensus that simplex is not required.
Analysis	

(Honeywell) I am having difficulty developing a justification for simplex mode, particularly if half-duplex mode is also part of the standard. If there is no half-duplex mode, then a simplex mode would allow a SpaceWire network to connect to simplex sources/sinks with some form of automatic detection (the simplex mode proposal I found did not include automatic detection, so I don't understand the need for the complexity described). Clearly there is no reason for greater use of simplex links. Since half-duplex mode would provide capability equivalent to simplex mode when used in a unidirectional manner and have all of the same SWaP benefits, an inability to implement half-duplex mode appears to be the only reason for simplex mode.

(RUAG) I don't understand the advantage of this mode compared to other links that are designed for simplex mode originally.

(TAS) I don't like the proposed solution for the lack of the flow control (as foreseen by the ECSS-E-50-12A – this seems also 4Links' opinion). A large buffer in the receiver is a cost (a router shall provide one buffer per port). In addition, once the size of this buffer is fixed and implemented in a device, a deep analysis shall be performed to highlight the effects of an overflow for any given application using that device. In our experience flow control ensures, besides to the consistency of the data exchanged through the SpW network, also the consistency of the operation at unit level, though in case of blocking it may cause data to be discarded at a source node generating data at high rate (e.g. an A/D converter).

(ASTRIUM) Router is not interesting for a simplex link, in case of point to point only link (w/o routing) then dedicated protocol is more efficient. Moreover the rationale to use simplex is not so simple since a camera for example need to be configured or initialized so it exists an up-link to the camera, (e.g. 1553) The harness mass reduction removing 1553 could be more important than reducing the SpaceWire harness except in half-duplex implementation where the configuration could be handled by the SpW link.

2.5 Requirements for Half-Duplex

2.5.1 Interoperability

Source	Proposal/response to ITT
Requirement	(Complete) interoperability with implementations that comply with ECSS-E ST-50-12C
Analysis	
This is dependent on the definition of interoperability. 4Links achieved the interoperability with autonomous detection that the link was operating in half-duplex, but do not wish to commit to this for the project. We do commit to achieving the requirement, but it may be via a hardware or software controlled switch rather than autonomous.	

2.5.1.1 Offered Definition of Interoperability

Source	**4Links**
Offered Definition	A device built to SpaceWire 1.1 1. may generate traffic that is ignored by a device built to ECSS-E-ST-50-12C; 2. may generate (as part of an initialization or recovery procedure) traffic that causes a device built to ECSS-E-ST-50-12C to disconnect, but shall not, except in such an initialization or recovery procedure, generate such traffic. 3. shall not generate traffic that a device built to ECSS-E-ST-50-12C will interpret as valid traffic but with different semantics from SpaceWire 1.1.
Analysis	
(4Links) The offered definition is specific to SpaceWire Evolutions and would be better if made totally generic to SpaceWire standards. The wording may not be adequately precise and needs to be considered carefully. (Astrium) “Totally agree” with this reservation. Also comment: (Astrium) The compliance definition is not clear since the implementation will have some limits. (4Links) Agree entirely with the lack of clarity of the suggested definition. (Astrium) A compliance test set should be defined for ECSS which allows to assess compliance of a device. The same for SpaceWire 1.1. (4Links) If there is a general ECSS definition of Interoperability which covers differences 1. within the Normative clauses of a standard? 2. between Normative and Optional clauses of a standard? 3. between Normative and Reserved features of a standard? 4. between different versions of a standard ? or if there is a definition in other standards, elsewhere than ECSS, that we could use, it might be preferable to inventing our own	

2.5.2 Generic requirements for half-duplex

2.5.2.1 Bidirectional communication

Source	Honeywell, RUAG, TAS, Astrium
Requirement	Half-duplex shall provide a communication path for both directions
Analysis	
(4Links) This requirement follows from the users’ rejection of Simplex in favour of Half-Duplex.	

2.5.2.2 Minimal change to the ECSS SpaceWire standard

Source	Proposal/response to ITT
Requirement	There shall be changes to as few sections of the ECSS-E ST-50-12C standard as possible
Analysis	
In 4Links' demonstrated implementation of half-duplex, no change was be required to: a: the Application level b: the Network level c: the Packet level d: the Character level Corresponding changes were required to: a: the Exchange level b: the Physical level Change may be required to: a: the Signal level If the offered definition of Interoperability above is accepted, an addition may be required to the Character level, but only preceding, and perhaps during, the initialization sequence.	

2.5.2.3 Half-duplex termination

Source	Honeywell, RUAG, TAS
Requirement	A specification of the permissible transmission-line terminations for the LVDS signals shall be defined in the SpaceWire 1.1 standard.
Analysis	
(TAS) How is each twisted pair terminated on each end ? 100 Ohm when receiving and open when transmitted, isn't it ? will be this point added to the ECSS ? A drawback is that available [LVDS-buffer] devices cannot support it (RUAG) The discussion about turnaround is good but I'm worried about the idea of two drivers driving simultaneously, even if it is only for a short time. In case of ground potential differences between the two nodes strange things may happen in an LVDS driver that is not designed for parallel driving. True bidirectional LVDS drivers, like for IEEE-1394 are not available for space use and since we already have a problem of funding the circuits we definitely need I would try to write the requirements such that standard LVDS circuitry can be used. (Honeywell) For half-duplex mode, the most immediate concern is the bidirectional connection used by a half-duplex link. Since LVDS links are destination terminated with 100 ohms, there clearly is an impact on the termination scheme that isn't addressed by the requirements. (4Links) Terminations are at both ends all the time, which is the bus LVDS standard. (4Links) Both ends driving simultaneously can only occur at start-up, and then for a short time. (4Links) Aeroflex do have a space-suitable bus LVDS driver (UT54LVDM031LV). Driving LVDS from 2.5V differential logic signals and a series/shunt resistor combination is safe from driving at both ends and is also safer with respect to propagating power-supply faults. (4Links) Although the specification of the transmitted signal and of the receiver termination can be identical to ECSS-E ST-50-12C, there needs to be text in the SpaceWire 1.1 standard that explains the termination and the potential occurrence of simultaneous transmission.	

2.5.2.4 Avoid simultaneous transmission of LVDS signals from both ends of the link

Source	**RUAG**

Requirement	The SpaceWire 1.1 standard shall either prevent simultaneous transmission from both ends of a link or, if prevention is not possible, set limits on the duration and extent of simultaneous transmission
Analysis	
(RUAG) ... I'm worried about the idea of two drivers driving simultaneously, even if it is only for a short time. In case of ground potential differences between the two nodes strange things may happen in an LVDS driver that is not designed for parallel driving. (4Links) This is only a problem at startup, and we believe that it may be possible to design a mechanism that avoids any such a condition at startup --- but it may require a small addition to the character level for activity on the link preceding startup. Crossover cabling and only transmitting on the Strobe differential pair connected at the other end to the Data differential pair (full-duplex pins) during initialization, for example, could ensure that there is never simultaneous driving.	

2.5.2.5 Priority over normal packet transfers to pre-empt blocked links

Source	Honeywell
Requirement	The turnaround protocol shall be fair
Analysis	
(Honeywell) I haven't found a detailed proposal for a half-duplex mode implementation (the 4-links presentation doesn't address link turn-around sufficiently). I am dubious that simply waiting for the current transmitter to run out of non-NULL characters is acceptable from a fairness perspective. Periodically changing direction to give the other end an opportunity seems like a straightforward way to address the fairness issue. There would be a slight performance impact if the other end immediately changed the direction back because it had nothing to send. A better solution would adjust the time interval for changing direction within some defined range (the dominant transmitter could eventually reach the maximum interval if the other end always returned the link, but would have to reduce the interval, possibly to the minimum time, if the other end transmitted data). (4Links) The currently transmitting end can only send a maximum of 56 Bytes before it runs out of flow-control and so has to send a Null. This ensures a periodic change of direction. Furthermore, if one end was concerned about giving up the link to the other direction for too long, it could limit the number of FCTs issued. We believe this is fair, but need a definition of fairness.	

2.5.2.6 Placeholder definition of fairness

Source	
Definition of fairness in the context of half-duplex turnaround	TBD
Analysis	
See the requirement that the turnaround protocol is fair.	

2.5.2.7 Robustness in the presence of network failures

Source	Astrium
Requirement	(Astrium) The priority of FCT or data to be transmitted first must be defined.
Analysis	
(Astrium) The priority of FCT or data to be transmitted first must be defined. (4Links) As the connection is only one direction at a time, the sender has to send both data and flow-control before handing over to the other direction. The best time to send the Flow Control information is as late as possible, i.e. after the data, but if implementers choose to do otherwise, must we prevent them?	

2.5.2.8 EMC consideration

Source	**RUAG**
Requirement	The project shall consider the EMC issues associated with enabling and disabling transmitters.
Analysis	
(RUAG) EMC: There is also an EMC aspect. LVDS is normally very good in terms of EMC aspects but with half duplex we will suddenly start to enable and disable the drivers at rather high frequency (especially when both nodes have nothing to send). This will cause current variations both inside the driving unit and on the link and the frequency of these variations may not be well controlled. This may then cause unwanted emissions in the UHF band (around 400 MHz) where several missions have very tight emission requirements due to Search and Rescue receivers or Proximity Links (for interplanetary missions). (4Links) There are possible EMC implications, probably mostly when the link is idle in both directions and so the direction is changing frequently. The effect of disabling a transmitter and leaving the line temporarily undriven is a signal half the level of a normal transition – it will not be worse than normal operation. RD15 suggests a turnaround time of 500ns which means that there is at least 1µs for a cycle of both ends turning round. A small variation in the turnaround time as suggested by Honeywell may mitigate EMC to some extent. 4Links' paper on EMC has some additional suggestions.	

2.5.3 Performance Requirements

2.5.3.1 Time Code and Interrupt Latency

Source	**Honeywell, RUAG, TAS, Astrium**
Requirement	The project shall consider specifying a maximum latency and a maximum jitter for Time Codes and for Distributed Interrupts.
Analysis	
(Honeywell) Another factor that is raised in some of the 4-links material (but not mentioned in the requirements) is the impact on time-code (et. al.) distribution latency and jitter due the oscillatory nature of the link direction. (RUAG) The performance aspects of Time Codes is relevant but what about the aspects when combining half duplex and Interrupt Codes? Here my concerns about unsynchronized timers etc. are probably even more relevant. (TAS) I understand the intrinsic increase of latency while forwarding time-codes (interrupt codes and packets) which could be compensated increasing the link rate (if possible) (Astrium) the use of half-duplex with the interrupt message have an impact on the interrupt acknowledge time-out and must be added in the time-out delay computation. (4Links) RD15 details a latency at 10Mb/s of nearly 62µs with a maximum of 7 flow-control tokens and 129µs for a maximum of 15 flow-control tokens. At 200Mb/s, the corresponding figures are 4 µs and 7.4µs respectively. If there are several routing switches in the path, this figure could be multiplied by the number of routing switches. And worst case with many interrupts will certainly not reduce the jitter. 4Links' paper on Reducing Time Code Jitter offers a means of reducing the effective jitter, but many systems may find that the above figures are acceptable.	

2.5.3.2 Throughput.

Source	Honeywell
Requirement	Throughput achieved should be no more than 10% worse than the figures reported in RD15
Analysis	

(Honeywell) You have identified performance requirements as missing and I agree that they are important, particularly with regard to router propagation.

(4Links) Router propagation will normally be a function of the link throughput. RD15 includes the following table of link throughput:

Bit rate Mbits/s	Unidirectional ½ duplex throughput	Bidirectional total ½ duplex throughput
10	7.3	7.5
20	14.4	14.8
50	34.3	36.1
100	63.6	69.3
200	111.4	128.7

It may not be possible, with other requirements placed on the project, to achieve exactly these figures, but an implementation goal of achieving no more than 10% worse than these figures is a reasonable requirement

2.5.3.3 Throughput trade-off with latency

Source	**4Links**

Request for consideration	Possible increase in Flit (Flow-control-unit) size, to improve throughput, but increase latency
Analysis	
(4Links) Flow-control unit (Flit) size and throughput: Throughput, especially at higher bit-rates, would be improved if more data was transferred before turnaround is required. This could be achieved by increasing the flow-control unit beyond 8-bits or by allowing more than 7 outstanding flow- control tokens. For example, if 15 flow- control tokens were allowed, the unidirectional throughput at 200Mb/s would rise from 111 to 130Mb/s and the bidirectional rate rise from 128 to 140Mb/s – see Table 4 (of RD15). A 17% increase in unidirectional rate is significant enough to be worth serious consideration at higher bit rates. (4Links) Increasing throughput necessarily increases worst-case latency. It is possible that for half-duplex protocols such as RMAP and SpaceWire-D, this does not matter, because the two directions will never compete. The maximum flow-control credit could be a function of link speed – this would limit latency and jitter, and would allow more efficient use of the link.	

2.5.4 Robust design

2.5.4.1 Flow-control without persistent state

Source	4Links
Request for consideration	Use a flow-control mechanism that does not rely on persistent state
Analysis	
(4Links) Half-duplex provides an opportunity to send an absolute value of the receive buffer-space available, which offers self-correcting flow-control. Comments are invited on whether this would be useful. **(RUAG) Concerning the self-correcting flow control it is not clear how that shall be implemented, unless you use the turnaround event as a special precondition to send the absolute value of the receive buffer space available instead of just sending FCTs. In our experience credit count errors basically never happens and we have had links running for years in space without any restart we know about.** **(4Links) Sending absolute values is not possible for full-duplex SpaceWire, but is possible with half-duplex because only one end transmits at any time.** **The suggestion was to send the absolute value of buffer space available.** **Most of the European flight experience so far has been with the original (1355) SMCS chips, whose core was designed by INMOS. That core was not perfect, but INMOS understood the asynchronous issues --- which many designers of SpaceWire have not. So RUAG's good experience may not be relevant to SpaceWire.** **Loss of flow-control credit has indeed been reported with SpaceWire – although it was subsequently found that the loss could have been avoided with a different design.**	

2.5.5 Physical layer

2.5.5.1 Connector type

Source	TAS
Request	Which type of connector will be recommended?
Analysis	

Source	TAS
Request	Which type of connector will be recommended?
Analysis	
(4Links) The existing Micro-D connector is suggested because that gives backwards compatibility and potential physical interoperability. If the NASA/Sabritec quadrax connector is included in the revision to the SpaceWire standard, it may be suggested as well. It is not clear that defining new connector or cable is within the scope of the project, but it is clear that this topic would be informed by what is being done on other ESA projects such as reduced mass cable. This request is combined with other Requests into Requirements 2.5.5.6 and 2.5.5.7	

2.5.5.2 Connector pinout

Source	Astrium
Request	The connector pinout for Micro-D must be defined
Analysis	
If a port is able to handle both mode the pin-out should be defined (which pins are half-duplex ready). It is important since symmetry is lost. I fear some interconnection mistake during assembly with standard and half duplex cable if we use the same connector. (4Links) The pinout is important and it is important to both preserve symmetry and to maximize compatibility with existing equipment. . This request is combined with other Requests into Requirements 2.5.5.6 and 2.5.5.7	

2.5.5.3 Bulkhead connection

Source	TAS
Request	Bulkhead connections?
Analysis	
(TAS) ECSS-50-12A doesn't consider the case of a cable connecting two units located in different parts of a platform separated by a barrier (this occurs on Ground too during thermal-vacuum test with the unit inside the TV chamber and the test equipment outside) as the two intermediate 9-pin SpW connectors on the barrier suffer the lack of 1 contact to provide continuity of the two shields tight to signal GND of one of the two ends. It would be convenient to standardise also matched intermediate connectors. (4Links) This request may also be under consideration in the Reduced Mass Cable project, and information about it would assist the SpaceWire Evolutions project.. This request is combined with other Requests into Requirements 2.5.5.6 and 2.5.5.7	

2.5.5.4 Cable type and specification

Source	RUAG, Astrium
Requests	Which type of cable will be recommended/specified?
Analysis	

Source	RUAG, Astrium
Requests	Which type of cable will be recommended/specified?
Analysis	
4Links used, as an example in RD15, some star-quad or Quadrax cable designed for space-flight use of IEEE 1394. (RUAG) First of all the cable comparison [in RD15] is a bit unfair since with a proper cable a half-duplex cable can only be 50% lighter than a full duplex cable (remember that the current standard cable is really overkill and could easily be replaced by two half duplex cables in parallel). This means that there cannot be any gain in bidirectional traffic and thus table 5 in the paper [RD15] is a bit misleading. (4Links) Agree with RUAG, but the Reduced Mass Cable project has not reported that it has considered this option, and its project leader did not appear to understand the benefits when they were described during WG16, so there is no certainty that the half-duplex cable will be used by that project. It is clear that the SpaceWire Evolutions project is not expected to deliver flight cables, but the project would certainly benefit from open access to information from the Reduced Mass Cable project, together with access to the silver-plated aluminium technology that they have said is being used. (Astrium) The maximum cable length versus data rate should be documented. It is necessary especially in launcher where long lengths are current. (4Links) Agreed. This request is combined with other Requests into Requirements 2.5.5.6 and 2.5.5.7	

2.5.5.5 Maximum distance for a given link speed

Source	Astrium
Request	Maximum distance for a given link speed should be defined
Analysis	
(4Links) This should also be defined for the existing SpaceWire cables. This request is combined with other Requests into Requirements 2.5.5.6 and 2.5.5.7	

2.5.5.6 Access to information from other ESA projects

Source	Requests xxx
Requirement	Access to technical information from other ESA projects
Analysis	
While not expected to deliver flight connectors and cables, the SpaceWire Evolutions project needs access to information and technology from other ESA projects, particularly other revisions to the SpaceWire standard and the Reduced Mass Cable project.	

2.5.5.7 Clarification of specification of physical layer

Source	Requests xxx
Requirement	Clarification of specification of physical layer for half-duplex, with respect to: Connector(s) Cable Treatment of bulkheads Connector pinout Distance/speed trade-off
Analysis	
The requests on connector type, cable type, bulkhead connectors, connector pinout and distance/speed trade-off are all valid, but it is not clear that they are within the scope of the project. This should be clarified with ESA.	

3 Technical Solution for Half Duplex SpW

3.1 Functional issues of Half Duplex SpW and proposed solutions

Full Duplex SpW was designed on the basis that two different and independent directions for the flow of data exist on the link. In order to support Half-Duplex SpW several modifications are required on the original standard.

The main challenges with Half Duplex SpW are the link initialization, the link direction reversal, the electrical standard which shall be adopted and the fact that the connector shall be redefined in order to result in greater mass reduction gain.

3.1.1 Physical and Signal Levels

The Full Duplex SpW signal level used Low Voltage Differential Signalling. LVDS technology is used for unidirectional point-to-point links, is not an appropriate solution for the implementation of Half-Duplex SpW and is not recommended for the implementation of the Half Duplex SpW signal level as is explained in this paragraph.

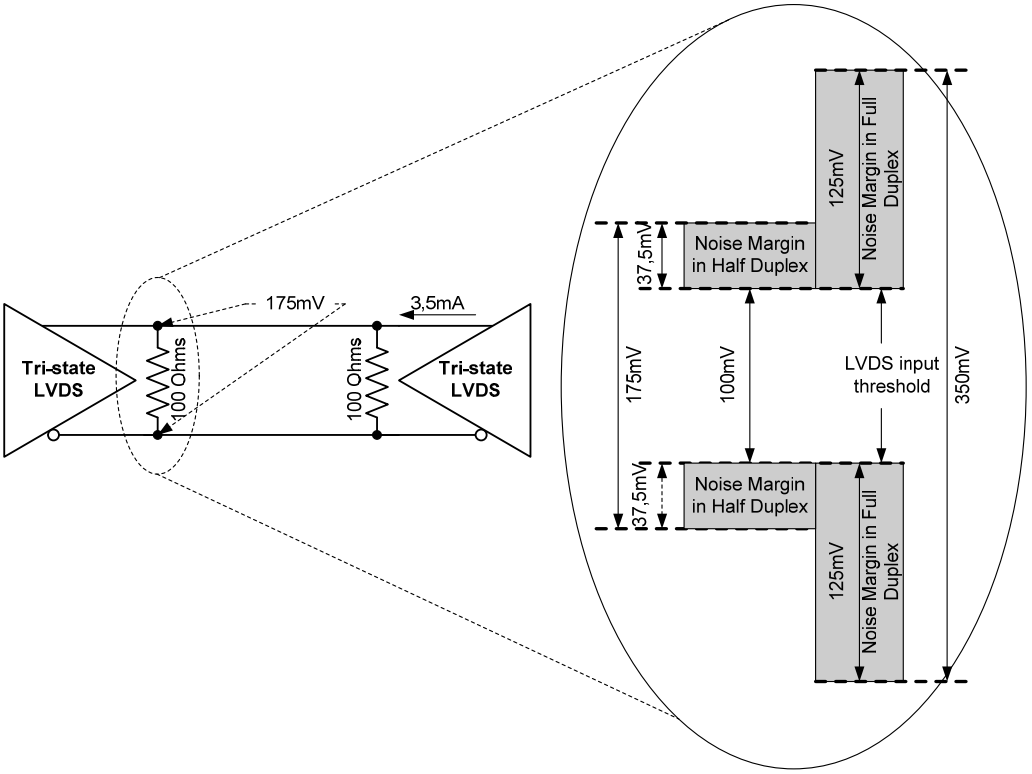

Figure 6: Noise margin reduction in Half Duplex SpW

LVDS drives resistor-terminated lines at high speeds. The LVDS transmitters drive a current of 3,5 mA and at the remote end of the line (receiver side) a voltage of 350 mV is developed at a termination resistor. Since LVDS receivers have a differential threshold of ±100 mV, this leaves another ±125 mV as noise margin.

In Half-Duplex SpW, the line shall be terminated at both ends, which means that the overall resistance in the link is 50 Ohms. Adopting LVDS for the implementation of Half-Duplex SpW means that the voltage at the input of each transceiver will drop to 175 mV, which lowers the noise margin ±37,5 mV as shown in Figure 6. This means that the BER will worsen and the EMC behaviour shall be re-evaluated.

Candidate technologies for the implementation of the Half-Duplex SpW signal level are the BLVDS and MLVDS. Their main features are summarized in Table 1.

LVDS (-A) (TIA/EIA 644 (-A))		Candidate technologies for Half-Duplex SpW Signal Level	
		M-LVDS (TIA/EIA 899)	**BLVDS (not standardized)**
Offset Voltage	1,125 – 1,375 V	0,3 – 2,1 V	1,185 – 1,435 V
Vout	454 mV (on 100 Ohms)	565 mV (50 Ohms)	350 (50 Ohms)
Transition time	260 ps	1000 ps	350 – 1000 ps
Driver strength	3,5 mA	11,3 mA	7 – 11,1 mA
Ground potential difference	±1 V	±2 V	±1 V
Input Voltage Range	0 – 2,4 V	-1,4 – 3,8 V	0 – 2,4 V
Input threshold	±100 mV	±50[1] mV	±100 mV
Max data rate (theoretical)	1,923 Gbps	500 Mbps	800 Mbps
Drivers contention	Not supported	Output current control	Output current control
Space Qualified Devices	Exist	Aeroflex UT54LVDM055LV	Aeroflex UT54LVDM031LV
Output voltage on 100 Ohms load	350 mV	1130 mV	700 mV – 1110 mV
Compatibility with LVDS		Analysis per design is required, Current at the receiver termination resistor may develop voltage more than 1 V.	

Table 1: Comparison of candidate technologies for Half-Duplex SpW Signal Level

Both technologies drive more current on the link that LVDS and therefore the EMC behaviour is different that the respective of LVDS. This practically means that a new cable definition will probably be required in order to meet the EMC requirements. To this respect the current study does not attempt to standardize the Signal or the Physical level of Half-Duplex SpW but will be restricted only to contributions based on the results obtained by experimentation on the prototypes that will be developed. Although experimentation is at an early stage the preliminary results have shown that:

- MLVDS and BLVDS commercial devices present good interoperability
- When the link is not driven by any of the link ends, the B/M LVDS transceivers "prefer" to park their receiver outputs at HIGH level. In most of the tests when the link was left floating by both ends a spike appeared which was eliminated (Figure 7) when the polarity of the LVDS signals was inverted. This may mean that in order to achieve error-free operation after the transmission of a NULL (when the link is not driven by any of the links in order to change the direction of transmission) the polarity of Half-Duplex SpW signals may need to be opposite than the polarity in Full Duplex SpW.

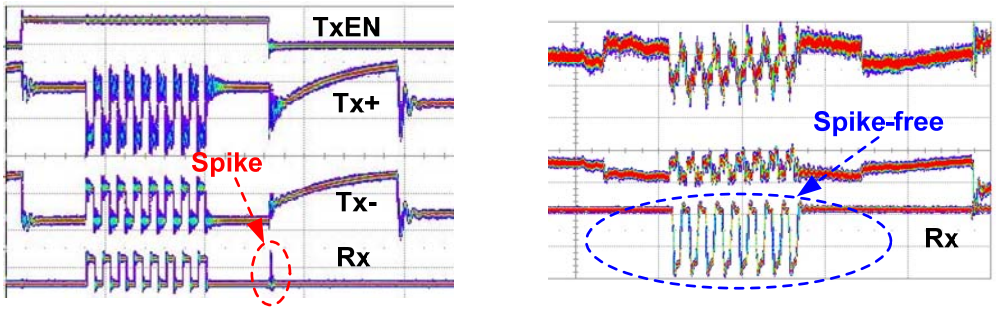

Figure 7: Spike elimination at the receiver output by signal polarity reversal

[1] MLVDS is standardized, whereas BLVDS is not. To this respect it would make more sense to adopt MLVDS for Half-Duplex SpW. However, the decreased input threshold shall be evaluated for EMC susceptibility.

3.1.2 Character Level

Many SpaceWire receivers extract the transmission clock by XORing the D-S pair. This has the following implications with Half Duplex SpW:

- After the transmission of a NULL bot the D and S signals do not toggle any more as shown in Figure 8.
- A SpW receiver is based on D-S transitions in order to extract the remote end transmission clock and decode the received characters
- This means that NULL detection shall be done asynchronously

Figure 8 also shows a potential implementation of a SpW receiver stage. The characters extracted from the link are shifter into a DDR shift register and a NULL decoder detects if a NULL has been received. The output of the decoder is sampled with the system clock in order to synchronize the signal to the state machine clock domain.

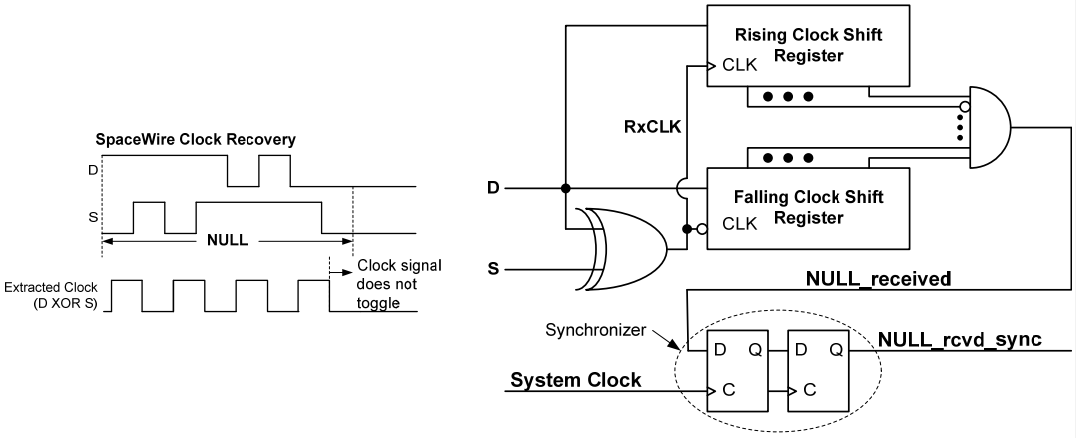

Figure 8: NULL character and SpW receiver stage implementation

However since no clock transition can be extracted by the D-S pair, the output of the decoder is not latched at the receiver's clock domain and is fed asynchronous to the state machine clock domain synchronizer. The output of the decoder may be momentarily asserted when the shift register shifts, due to differences in propagation delays.

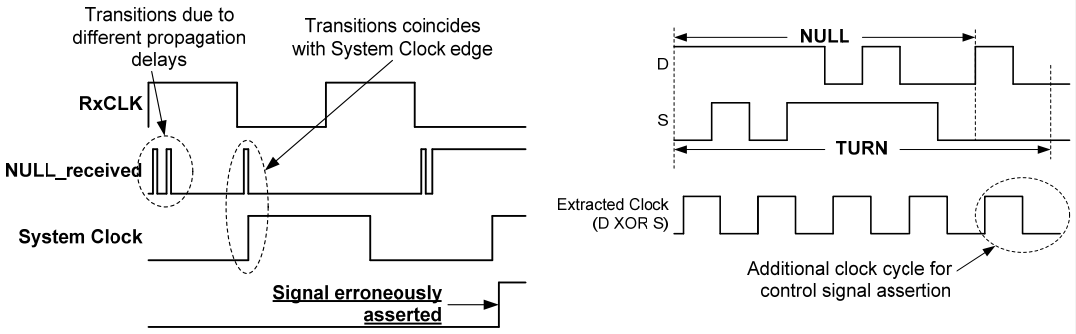

Figure 9: Erroneous NULL detection due to the absence of latch at the output of the NULL decoder and the TURN character

As shown in Figure 9 this may cause erroneous NULL detection if the system clock coincides with such a momentarily assertion and will cause erroneous link direction reversal. In order to overcome this situation, an additional transition on the D-S pair is required in order to allow latching the output of the decoder on the receiver's clock domain. For this reason the introduction of a new character (shown in Figure 9), the TURN character, is proposed for link direction reversal in Half-Duplex SpW.

3.1.3 Exchange Level

The State Machine of Full Duplex SpW ensures that the two ends of the link pass from the same states simultaneously (or almost simultaneously) during initialization. In case a controller starts up before the remote end has powered up the disconnect signal ensures that the controller will be reset and will repeat the cycle until the remote end has powered up. In case a non-expected character is received by the remote end, a disconnect is provoked to ensure that the two ends start again from the ErrorReset state and pass (almost) simultaneously from the same states.

This, however is not possible with Half Duplex SpW, because in such a case the two ends of the link will be driving the link or listen to it simultaneously. To this respect an asymmetry is inserted in the state machine to ensure that the two ends pass from different states during initialization and one end will be driving the link while the other will be listening to it.

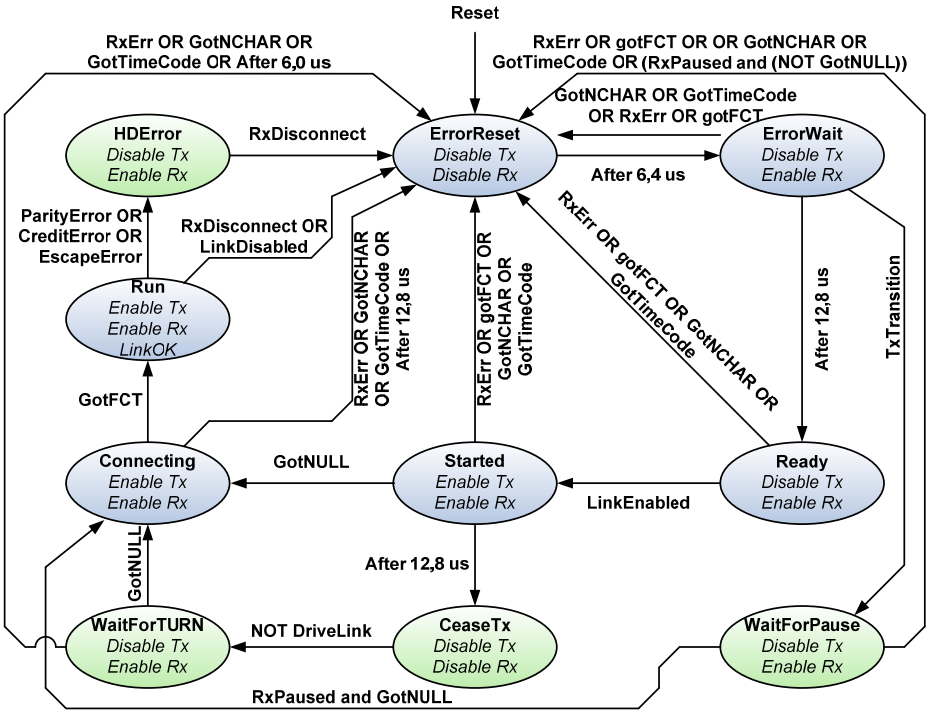

Figure 10: The Half Duplex SpW state machine

Compared to the Full Duplex state machine four new states and three new signals are introduced.

New signals:

- ***TxTransition***: Indicates that a transition has been detected on its Tx D-S pair. This is the main signal that causes the asymmetry in the operation of the two ends of the link. If end A detects a transition on the line, this means that the remote end has already driven the line so end A shall initialize in "listen" mode.
- ***RxPaused***: Indicates that no transition has been detected on the D-S pair for 200 ns. It signals that the remote end does not drive the link any more. It is used by the "listening" end to detect when the remote end has paused transmission.
- ***DriveLink***: Indicates that the link is driven by the controller's transmitter. Deasserted after the last character has been transmitted and the transmitter does not drive the link. It is used by the "driving" end and causes it to enter "listening" mode after it has sent all of its characters that it shall send until it reaches the ***Started*** state.

New states:

- ***WaitForPause***: Entered from the ***ErrorWait*** state if a transition has been detected on the transmit D-S pair. The state machine waits here until the remote end does not drive the line (***RxPaused***) any more.

- ***CeaseTx***: Entered from ***Started*** state if no NULL has been received for 12,8 us, a condition which may indicate that the remote end is a full duplex (or half duplex) and is disabled, or it is half duplex and waits its turn for transmission (or disabled).
- ***WaitForTURN***: Entered from the ***CeaseTx*** state. At this state the transmitter is disabled to allow the remote end to transmit its NULLs and FCTs
- ***HDError***: Entered from the ***Run*** state when a Flow Control or Parity error is detected. At this state the receiver waits for the transmitter to cease transmission before proceeding to the ***ErrorReset*** state so that the two ends will proceed with the "Exchange of Silence" mechanism to reinitialize the link.

3.1.3.1 Link Initialization

The functionality of the state machine presented in Figure 10 can be explained with the help of the sequence diagram shown in Figure 11 and explained in Table 2.

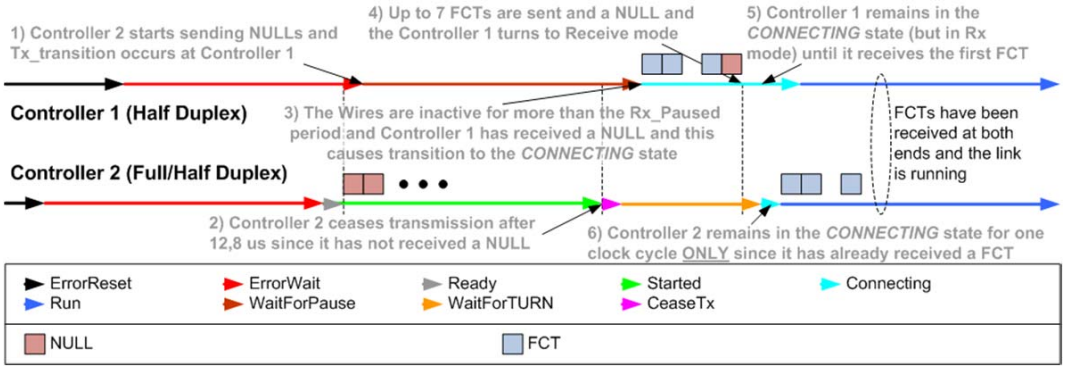

Figure 11: Half Duplex Link Initialization

Controller 1		Controller 2	
State	**Condition**	**State**	**Condition**
ErrorReset		***ErrorReset***	
ErrorReset		***ErrorReset***	The 6,4 us timer expires causing Controller 2 to enter the ***ErrorWait*** state
ErrorReset	The 6,4 us timer expires causing Controller 1 to enter the ***ErrorWait*** state	***ErrorWait***	
ErrorWait		***ErrorWait***	The 12,8 us timer expires at Controller 2 faster than the respective timer at Controller 1, and Controller 2 enters the ***Ready*** state
ErrorWait		***Ready***	***LinkEnabled*** is asserted causing Controller 2 to enter the ***Started*** State
ErrorWait	A transition is detected on the D-S pair (due to the transmitted NULL by Controller 2) and thus signal "***TxTransition***" is asserted causing Controller 2 to enter the ***WaitForPause*** state	***Started***	Controller 2 starts transmitting NULLs
WaitForPause	Controller 1 waits for Controller 2 to stop driving the link	***Started***	Controller 2 keeps transmitting NULLs
WaitForPause	Controller 1 waits for Controller 2 to stop driving the link	***Started***	The 12,8 us interval expires and it proceeds to the ***CeaseTx*** state

Controller 1		Controller 2	
State	**Condition**	**State**	**Condition**
WaitForPause	Controller 1 waits for Controller 2 to stop driving the link	***CeaseTx***	It ceases transmission and proceeds to the ***WaitForTURN*** state
WaitForPause	After 200 ns the pause_rx signal is asserted and since it has got a NULL it proceeds to the ***Connecting***	***WaitForTURN***	
Connecting	Controller 1 starts transmitting FCTs	***WaitForTURN***	
Connecting	Since it has not received FCTs from the remote side it transmits a NULL and returns to receive mode after having transmitted all its FCTs	***WaitForTURN***	
Connecting		***WaitForTURN***	Controller 2 receives the NULL from Controller 1 and proceeds to the ***Connecting*** state
Connecting		***Connecting***	Controller 2 has already received FCTs, so signal ***GotFCT*** is already asserted and its state machine proceeds to the ***Run*** state
Connecting		***Run***	Controller 2 transmits its FCTs
Connecting	Controller 1 receives the first FCT from Controller 2 and proceeds to the Run state	***Run***	
Run		***Run***	
The Half Duplex Link is initialized			

Table 2: Half Duplex State Machine transitions

3.1.3.2 The Half-Duplex SpW state machine livelock and mitigation methods

The state machine presented in the previous section offers link initialization in both Half Duplex and Full Duplex SpW and also offers the capability of Half/Full Duplex auto-detect. There is an improbable but possible situation in which the two ends of a link may enter a Livelock either temporarily or permanently, in case the two ends of the link are enabled simultaneously. This situation and mitigation methods are provided in this section.

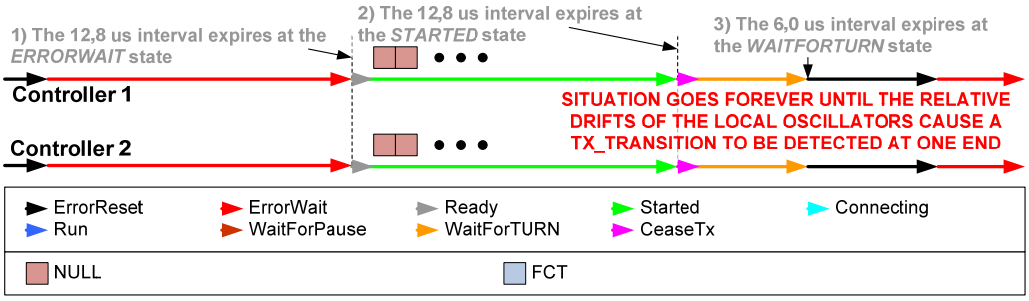

Figure 12: The Half Duplex SpW Livelock

The Livelock situation is shown with the help of Figure 12. In this example we assume that the two controllers happen to be simultaneously activated. What will happen next is that they will continuously be passing through the same states simultaneously which means that they will be both driving the bus at the same time or be listening to the bus at the same time.

Depending on the clocking scheme this may mean that initialization may never occur or may take a long time.

- In case the two controllers are clocked by different sources the initialization time will depend on the relative drifts of the respective oscillators
- In case the two controllers are clocked by the same source, initialization will never occur since there is no relative drift.

As shown in 3.1.3.1, in order to achieve initialization, at some point in time one of the controllers shall be at the ***ErrorWait*** state listening to the bus and the other one shall be at the ***Started*** state transmitting NULLs.

Assuming that Controller 1 enters the ***Started*** state and transmits a NULL, this means that at the remote side the first transition will occur after $t_{PROPAGATION}$, which is the propagation delay on the cable. This transition will not be detected at the receiver side immediately since the path to the State Machine will normally involve multi clock domains which impose 2-3 cycles delay of the system clock at the receiver side. This means that the first transition will be perceived after $t_{PROPAGATION}$ + 3xSystemClockPeriod$_2$.

In addition at the transmitter side, depending on the implementation, the NULL may not be transmitted immediately since the transmitter may also involve clock domain crossing. The worst case is that the controller uses the transmission clock (5MHz at initialization) to sample the command from the state machine which means that it will require 400ns – 600 ns (2 – 3 clock cycles) to synchronize the command from the state machine and enter the ***Started*** state. On top of this, upon activating the line buffer, it is not possible to start driving the line to logical '1' immediately, but an initial delay shall be inserted, so another 200 ns are added to this (1/5MHz).

Summing all of the above this means that in order to achieve initialization this means that Controller 1 shall enter the ***Started*** state before Controller 2 by:

$$600\text{ ns} + 200\text{ ns} + t_{PROPAGATION} + 3\text{xSystemClockPeriod}_2\text{.(Desired Drift Time)}$$

For a 10 meters cable and a clock frequency of 50MHz at the receiver this translates to approximately 1 us.

Assuming that the two controllers are clocked by two oscillators with frequencies F1 MHz and F2 MHz respectively this means that for every us

- Oscillator A requires F1 clock cycles
- Oscillator B requires F2 clock cycles

Assuming that the two oscillators have X and Y ppm accuracy, the best case to initialization will occur if Oscillator A oscillates at F1-X ppm and Oscillator B at F2 + Y ppm (or vice versa). This means that every 1/F1 seconds Oscillator A drifts negatively by X microseconds and every 1/F2 seconds Oscillator B drifts positively by Y microseconds. To simplify calculations in this example let's assume that F1 = F2 = F. This means that after 1/F seconds we have a relative drift of X+Y microseconds. This means that the desired drift will occur after 1/(F x (X+Y)). For a frequency of 50 MHz and X = Y = 20 this results in 1/2000 = 0,5 milliseconds and this is for the BEST case in which both oscillators pull towards different directions and at maximum deviation.

This situation becomes even worse in case the two ends do not from the same states at the same exact time, but one of the ends (the slower) is has just entered the ***Started*** state and the other end (the faster) will enter the ***Started*** state in 1 us.

In order to mitigate the Livelock problem and shorten the worst-case initialization time, an asymmetry shall be introduced in the link. This section analyses potential solutions for two configurations and concludes with a proposed solution.

In case the two connected ports have a common clock source (e.g. two ports of the same device) there is no time drift and this situation perpetuates. In order to overcome this situation it is recommended that the SpW ports are "numbered" and each port has an offset to the SpW timers related to the port number. This means that each port timer will have a "unique" timer value and this situation should never occur. In the case of a router this is easy to implement, by assigning numbers through VHDL generics, however care shall be taken at system level design. If for example two SpW devices on the same PCB share their clock source this issue shall also be taken into account.

Connecting two ports of different devices does not result in perpetuated Livelock as above. However it may result in very long Link Initialization time. Different solutions are examined below:

- Add a pseudo-random offset at the 12,8 us timer. This is a solution like the one that is adopted by Ethernet. A pseudo-random generator shall add a positive or negative offset to the 12,8 us time each timer the timer runs. Even with this case care shall be taken at system level in order not to assign the same "seed" to the pseudo-random generator since otherwise the offsets that will be added to the timer will be the same and Link Initialization time will also be depend on the relative oscillator drifts.
- Add a fixed offsets at the two ends of the link. One side shall have a positive offset and the other one shall have a negative offset. For example, devices that are RTUs or Routers shall have a positive offset to the timers and devices that are end nodes shall have a negative offset (or vice

versa). Although this destroys the symmetry of SpW, it is anticipated that Half Duplex links will only be adopted for the boundary of a SpW network. The reason for this is the latency it inserts in packet propagation and time-code distribution which makes it more likely not to be used as a core network element.

- A solution similar to the above would be to configure end nodes in auto-start mode and Routers or RTUs in Link Enabled mode.

The method described for devices that share a common clock source is mandatory to be implemented in devices with more than one port.

For the methods described for devices with different clock sources, the first one is the most complicated in terms of hardware implementation and also requires different seeds at the two ends of the link. Since not all SpW devices have embedded processors, this may require the existence of external strap pins increasing cost and complexity at PCB level.

Adding a fixed positive/negative offset at the two ends of the links is easily implementable no matter if the node has an embedded processor or not. This scheme is expected to shorten the link initialization time in links in which one end is a Router/RTU and the other end is an end node, but it will not have any impact in point-to-point links between end nodes.

Finally configuration of one of the links in auto-start mode is also easily implementable and provides a solution in point-to-point links between end nodes.

Summarizing all of the above it seems that the best solution would be to:

1. Add a port number VHDL generic at SpW CODEC level, which will add different offsets on the SpW timers for each SpW port. This approach provides solution in systems with common clock source whose ports are connected to each other.
2. Provide, at SpW CODEC level, the capability to force the offset to positive or negative direction. Positive will be used for Routers/RTUs and negative for end nodes (or vice versa). Activation of the feature should be selectable so that it can be used only if desired.
3. Provide, at SpW CODEC level, auto-start mode functionality (already provided for Full Duplex so no interface modifications are foreseen). This will also be selectable by the user and should be used in point-to-point node links, if desired.

The combination of the three techniques proposed above will shorten Link Initialization time in all possible Half Duplex configurations. In any case however, whether the mitigation method will be adopted or not, deterministic Link Initialization time cannot be guaranteed.

3.2 Half Duplex SpW performance

In this section performance aspects of Half Duplex SpaceWire are analysed.

3.2.1 Half Duplex SpW Latency

With Half-Duplex SpW a node (or router) having data to transmit may not be able to transmit them immediately since the remote end of the line may be driving the line. Therefore it shall wait until the remote end has sent all its data and the TURN character before transmitting its data. This inserts latency in the transmission of SpW packets and Time-Codes which depends on the amount of data the remote end has to send, on the link speed and on the time needed for the direction reversal.

The worst case scenario is the one in which on end is just releasing the link and the remote end is just possessing the link and has the maximum number in its credit buffer and data to send, which means that it will transmit all data for which it has credit and will also transmit the maximum number of FCTs because it has already consumed the received data. This means that in order to transmit the first NCHAR or Time-Code the node (or router) shall wait for:

- The turnaround time of the link for the remote end to start driving the link
- The transmission of the remote end FCTs
- The transmission of the remote end data (NCHARs)
- The turnaround time in order to start driving the bus
- The transmission of its own FCTs

The maximum number of data sent by the remote side is directionally proportional to the maximum number of FCTs. The impact of the maximum number of FCTs in the worst case latency is shown for different FCTs in Table 3 and Figure 13. As can be seen the latency increases from 870% up 18000% for a maximum

number of 20 FCTs[1] and this worst case may occur per link in some cases, which means that it is additive in cascaded Half-Duplex SpW paths.

Worst case latency, in microseconds) vs. maximum number of FCTs						
Link Speed (Mbps)	**Full Duplex**	**5 FCTs**	**7 FCTs**	**10 FCTs**	**15 FCTs**	**20 FCTs**
10	1	43,3	60,1	85,3	127,3	169,3
50	0,2	9,06	12,42	17,46	25,86	34,26
100	0,1	4,78	6,46	8,98	13,18	17,38
150	0,067	3,35	4,47	6,15	8,95	11,75
200	0,05	2,64	3,48	4,74	6,84	8,94
300	0,033	1,92	2,48	3,32	4,72	6,12

Table 3: Worst case latency in the transmission of NCHARs and Time-Codes vs. maximum number of FCTs

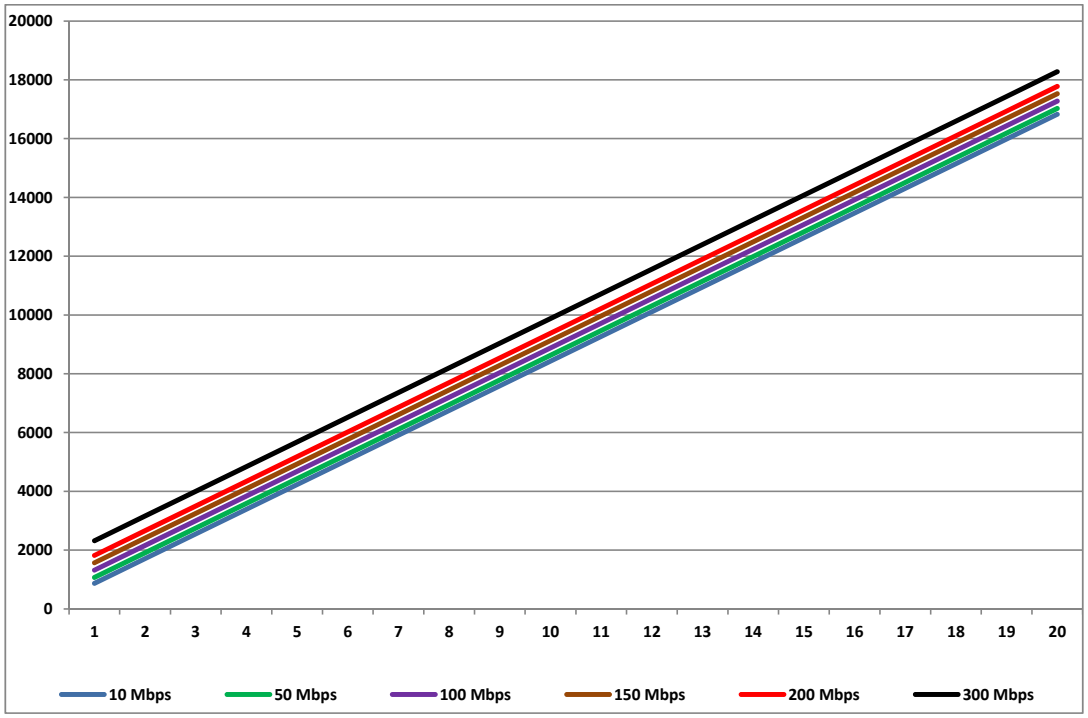

Figure 13: Impact of the maximum number of FCTs on latency as a percentage

3.2.2 Half Duplex SpW bandwidth utilization

From section 3.2.1 it becomes clear that the best latency is achieved for the minimum number of FCTs. However minimizing the number of FCTs means that there will be more turnaround cycles, since the credit are consumed earlier and each end shall pass the link possession to the remote side in order to get FCTs and refresh its credit. In this section the impact of FCTs and link turnaround time on bandwidth utilization is analysed.

[1] Although 20 FCTs are non-compliant with SpW, this is examined, because (as will be seen in 3.2.2) it results in better bandwidth utilization

3.2.2.1 Bandwidth utilization vs. FCTs

Reducing the maximum number of FCTs means that there will be more overhead for link direction reversals. The impact on bandwidth utilization also depends on whether only one of the ends has data to send or if both of the ends have data. This is because in the first case there an overhead of two turnaround times for a chunk of data whereas in the second case there the overhead is one turnaround time. In addition, FCT from one side of the link are sent if the traffic is unidirectional. Both cases are examined in this section since the reason for initially proposing Half-Duplex SpW was for unidirectional data traffic.

Available bandwidth utilization for unidirectional traffic					
Link Speed (Mbps)	**1 FCT**	**5 FCTs**	**10 FCTs**	**15 FCTs**	**20 FCTs**
10	73%	90%	92%	93%	94%
50	53%	82%	88%	90.5%	91.5%
100	40%	75%	83%	87%	89%
150	32%	68%	79.5%	84%	86%
200	27%	63%	76%	81%	84%
300	20%	54%	69%	77%	80%

Table 4: Bandwidth utilization vs. maximum number of FCTs for unidirectional traffic

Available bandwidth utilization for bidirectional traffic					
Link Speed (Mbps)	**1 FCT**	**5 FCTs**	**10 FCTs**	**15 FCTs**	**20 FCTs**
10	82.5%	92%	94%	94%	94.5%
50	68%	88%	91%	93%	93.5%
100	56%	83%	89%	91%	92%
150	48%	79%	86.5%	89%	90.5%
200	42%	76%	84%	87%	89%
300	33%	69%	80%	85%	87%

Table 5: Bandwidth utilization vs. maximum number of FCTs for bidirectional traffic

This impact is presented in Table 4 and Table 5 for unidirectional and bidirectional data traffic and also shown graphically in Figure 14.

From the tables and the figure it becomes apparent that:

- Decreasing the maximum number of FCTs decreases the Half-Duplex SpW bandwidth utilization
- The impact is more significant for high link speeds since the overhead inserted by the link turnaround time corresponds to more bits that could have been sent

Summarizing the results of this section and of section 3.2.1 it seems that a maximum number of 7 FCTs is a good trade-off, since it offers more than 75% bandwidth utilization and in addition it means that no modifications are necessary regarding the credit logic of the existing SpW IP Cores.

In addition to the above, Figure 15 shows the impact of FCTs on bandwidth utilization as a percentage. As can be seen the impact is significant for unidirectional traffic. Furthermore, in the case of unidirectional traffic, the latency for the transmission of data is not prohibitively increased since consists only of the turnaround time and on the time needed to exchange the FCTs, which are sent on one direction only. To this respect increasing the maximum number of FCTs for unidirectional applications would make sense if the increased cost for larger receive buffers justifies the gains shown above.

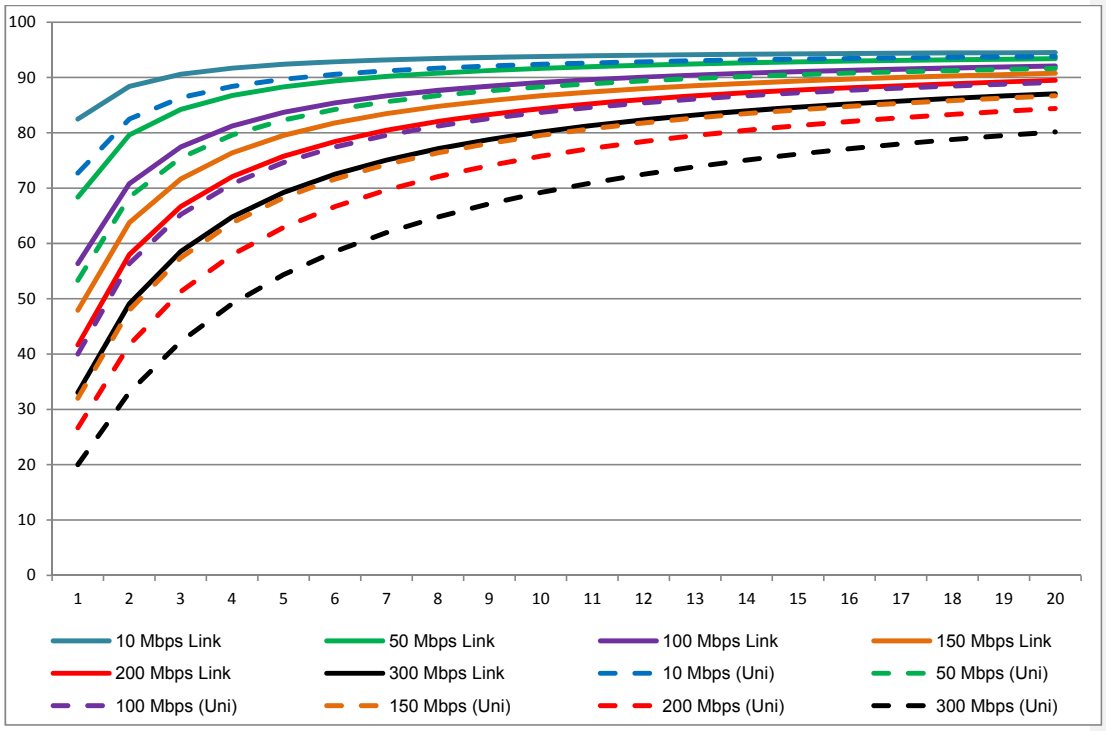

Figure 14: Maximum bandwidth utilization (percentage) vs maximum number of FCTs

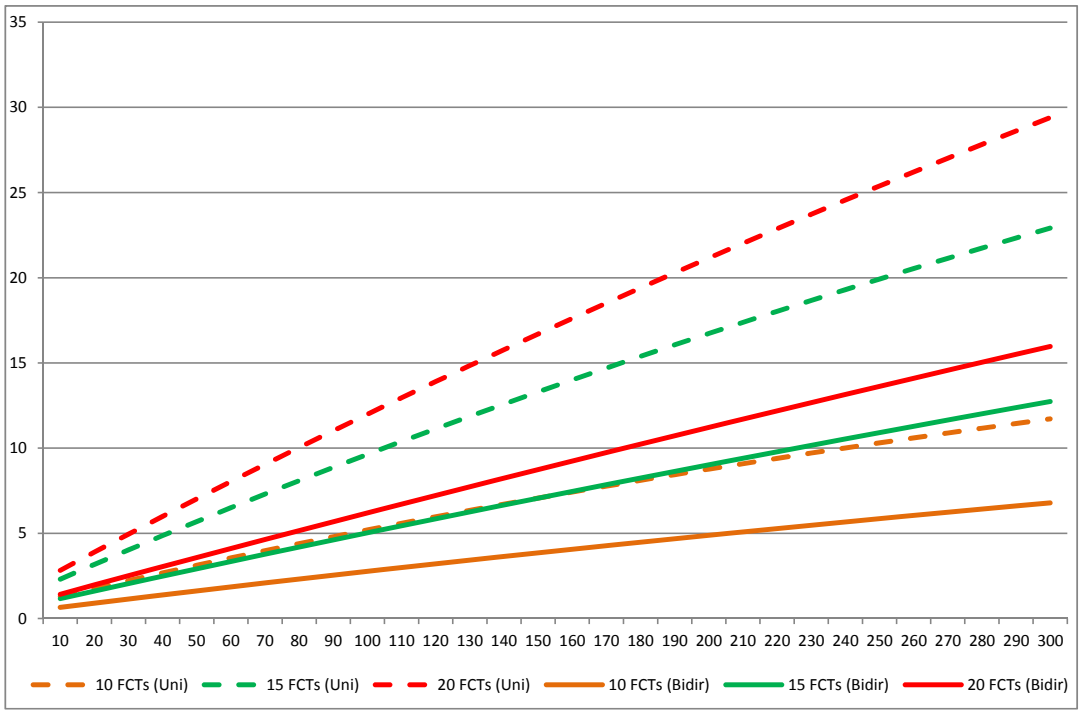

Figure 15: Impact of maximum number of FCTs on bandwidth utilization as a percentage

3.2.2.2 Bandwidth utilization vs. Link turnaround time

The link turnaround time has a slight impact on latency but a more severe impact on bandwidth utilization, especially at high link speeds since it is an overhead time consuming an amount of time which corresponds to a significant number of bit periods.

The maximum bandwidth utilization for unidirectional and bidirectional traffic for various link speeds and 7 FCTs is shown in Table 6 and Table 7 respectively and a graphical representation appears in Figure 16.

Available bandwidth utilization vs. turnaround time unidirectional traffic					
Link Speed (Mbps)	**200 ns**	**400 ns**	**600 ns**	**800 ns**	**1000 ns**
10	92%	91%	91%	90%	90%
50	91%	87%	84%	82%	79%
100	87%	82%	77%	73%	69%
150	84%	77%	71%	66%	62%
200	81%	73%	66%	60%	56%
300	77%	66%	58%	51%	46%

Table 6: Bandwidth utilization vs. turnaround time for unidirectional traffic

Available bandwidth utilization vs. turnaround time bidirectional traffic					
Link Speed (Mbps)	**200 ns**	**400 ns**	**600 ns**	**800 ns**	**1000 ns**
10	93%	93%	93%	92%	92%
50	92%	91%	90%	88%	87%
100	91%	88%	85%	83%	80%
150	89%	85%	81%	78%	75%
200	88%	83%	78%	74%	70%
300	85%	78%	72%	67%	62%

Table 7: Bandwidth utilization vs. turnaround time for bidirectional traffic

It becomes apparent that:

- The turnaround time has indeed a significant impact at high link speeds, in which the utilization drops to less than 75%
- The turnaround time has a more significant impact for unidirectional traffic flows, since the overhead introduced is double
- The bandwidth consumed by the turnaround time is inversely proportional to the maximum number of FCTs exchanged, since less turnarounds occur due to the increased credit level (shown in Figure 18)

Since the turnaround time is mainly required to compensate for cable propagation delays and provide relaxation time for the electrical signals on the link, which means that it related to the cable length, a proposal could be to start a Half Duplex link at a default value (e.g. 500 ns) and adjust it according to the cable length. This, however, is theoretical since it is based on the assumption that Half-Duplex SpW can indeed achieve link speeds of over 200Mbps which will be evaluated during the prototype implementation.

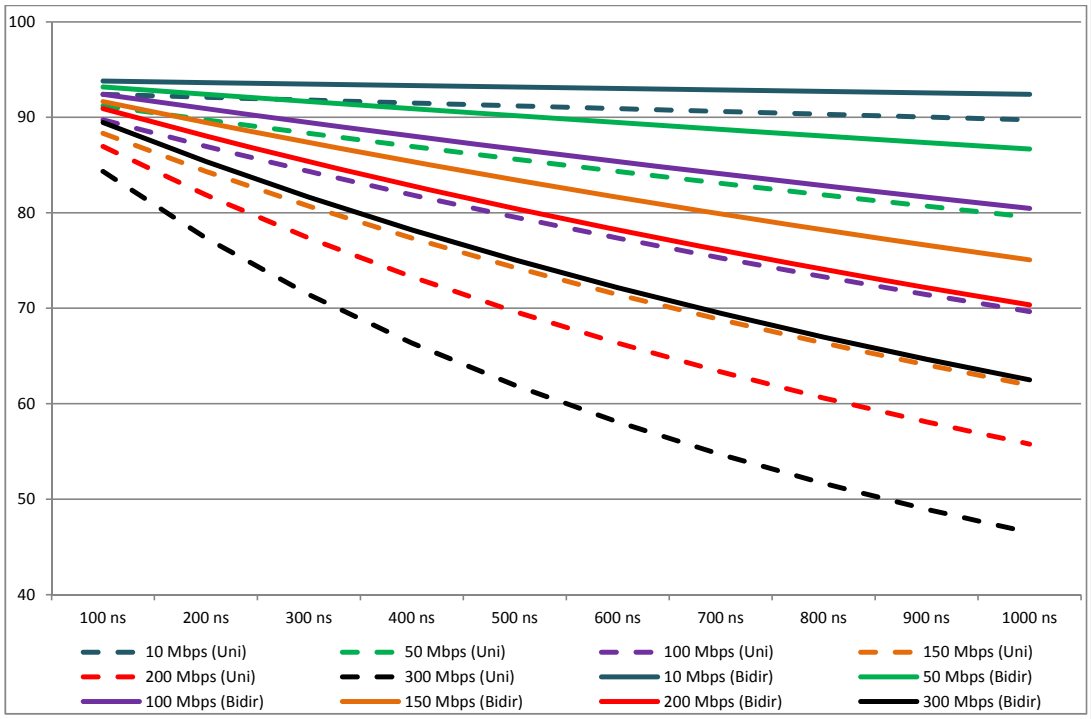

Figure 16: Maximum bandwidth utilization (percentage) vs. turnaround time for 7 FCTs

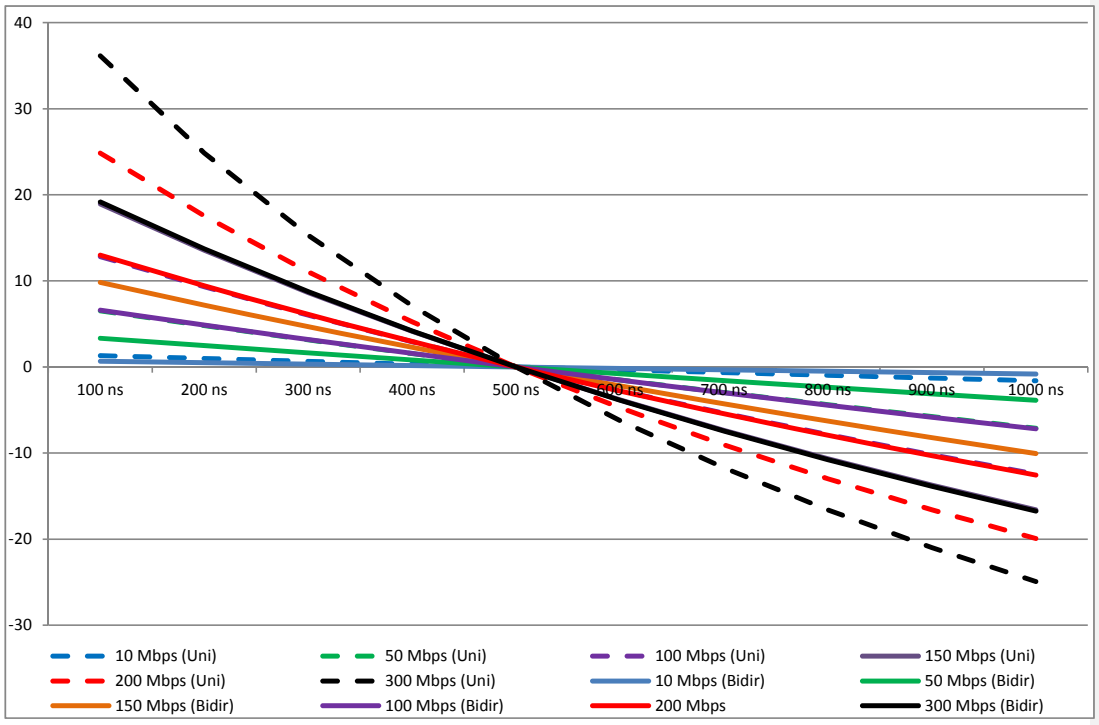

Figure 17: Impact (as percentage) of turnaround time on bandwidth utilization

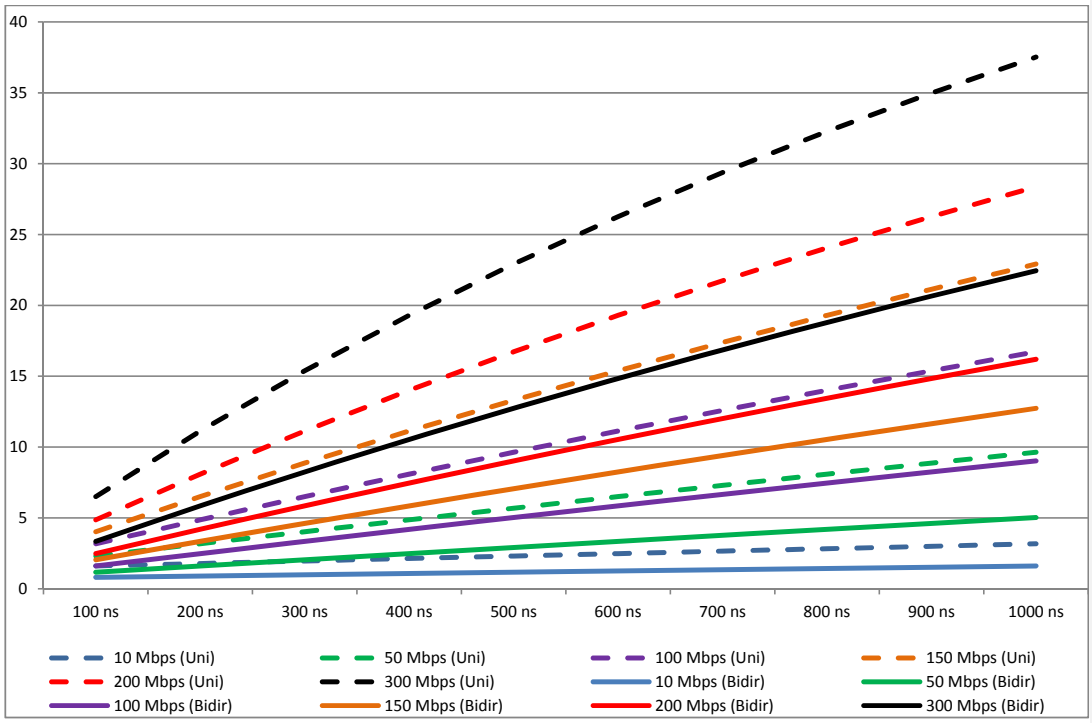

Figure 18: Bandwidth utilization increase (as percentage) vs. turnaround time by increasing the FCTs from 7 to 15

3.3 Conclusions

Half-Duplex SpaceWire is a promising solution for low mass SpaceWire since:

- ☑ It supports all SpW 1.0 functionality and does not infer the hazard of NCHARs loss as Simplex (discussed in)
- ☑ It supports wormhole routing
- ☑ It allocates bandwidth fairly among the two link ends
- ☑ The modifications required on the SpW implementations are restricted since most of the functionality is identical
- ☑ It is compatible with the Interrupts and PnP SpW Evolutions
- ☑ It can support scheduled communication, if the jitter requirements are not strict, or if the data traffic is unidirectional
- ☑ It requires simpler and lighter cable – this is TBC after EMC characterization
- ☑ It's a low cost solution for networks with few hops
- ☑ It' s, proposed, state machine is capable of auto-detecting whether the remote end of the link operates in Half or Full duplex mode

Although a promising solution for light mas SpaceWire, Half Duplex presents the following drawbacks:

- ☒ Physical level modification is required. A new connector should be defined to achieve optimized throughput vs. mass performance
- ☒ It requires modifications at the Signal Level in order to compensate the increased current requirement for the termination resistors at both line ends
- ☒ It has be EMC-characterized due to the increased current drive requirement
- ☒ It requires the introduction of a new character (TURN)
- ☒ It has increased Time-Code and NCHARs latency and jitter
- ☒ It cannot support precise time distribution

- ☒ It may present excessive jitter in hot redundant topologies, thus making copies filtering more difficult
- ☒ The efficiency and latency are factors driving towards opposite directions and trade-off analysis per design may be required

Given its pros and cons, Half-Duplex is more well suited for applications at the network boundaries (e.g. RTUs) which concentrate data and can propagate them to the core network isochronously through Full Duplex links. Adoption of Half Duplex in the core network is not recommended, except for application, without any timing constraints, since the latency in cascaded Half Duplex paths may be additive.