SpaceFibre

Steve Parkes¹, Chris McClements¹, Martin Dunstan¹
Albert Ferrer², Alberto Gonzalez²

¹Space Technology Centre, University of Dundee. UK
²STAR-Dundee Ltd, UK
Contents

- SpaceFibre Applications
- SpaceFibre Network Level
- SpaceFibre Standard Update
SpaceFibre Applications

Local Instrument
Data Output

Instrument 1 Interface

Instrument 2 Interface

Mass Memory Unit

Data Output

Data Bus
To Memory
SpaceFibre Applications

Local Instrument
- Data Output
  - Instrument 1 Interface

Local Instrument
- Data Output
  - Instrument 2 Interface

Local Instruments
- Data Output
  - SpaceWire To SpaceFibre Bridge

Remote Instruments
- SpaceWire Instrument
- SpaceWire Instrument
- SpaceWire Instrument
- SpaceWire Instrument

Mass Memory Unit
- Mass Memory Interface
  - Data Bus To Memory
SpaceFibre Applications

Local Instrument
Data Output
Instrument 1
Interface

Local Instrument
Data Output
Instrument 2
Interface

Local Instruments
Data Output
SpaceWire
To SpaceFibre
Bridge

Mass Memory Unit
Mass Memory
Interface

Downlink Telemetry
Downlink
Telemetry
Interface

Data Bus
To Memory

Remote Instruments
SpaceWire
Instrument
SpaceWire
Instrument
SpaceWire
Instrument
SpaceWire
Instrument
SpaceFibre Applications

Local Instruments
- Data Output
- SpW Control/HK

Remote Instruments
- SpW Control/HK

SpaceWire to SpaceFibre Bridge
- Data Output
- SpW Control/HK

Instrument 1 Interface
- Data Output
- SpW Control/HK

Instrument 2 Interface
- Data Output
- SpW Control/HK

SpaceWire Control/HK

SpaceFibre Router

Control Processor
- Data Input/Output
- SpW Control/HK

Mass Memory Unit
- Data Bus To Memory

Downlink Telemetry
- Data Output
- SpW Control/HK

Local Instrument
- Data Output
- SpW Control/HK

Control Processor Interface
- SpW Control/HK

Mass Memory Interface

Downlink Telemetry Interface
- SpW Control/HK
Principal On-board Communications

- **Instrument**
  - Data to mass-memory
  - Configuration and control
  - Housekeeping

- **Mass memory**
  - Data from instruments
  - Data to downlink
  - Configuration and control
  - Housekeeping

- **Downlink**
  - Data from mass-memory
  - Configuration and control
  - Housekeeping
SpaceFibre Network Level

- SpaceFibre transfers SpaceWire packets
  - Destination address
  - Cargo
  - EOP

- Path and logical addressing can be used

- Routing concept identical to SpaceWire

- VCs can be used to provide
  - Virtual networks – like SpaceWire
  - Constrained virtual networks
  - Virtual point to point links

- Improved FDIR
Virtual channel buffers are configured to support specific virtual channels. One set of buffers is always configured to support VC 0, the Configuration Virtual Network.
SpaceFibre Virtual Point to Point Link

- **SpFi Port 1**: VC 6, VC 2
- **SpFi Port 2**: VC 4, VC 6
- **SpFi Port 3**: VC 6, VC 4
- **SpFi Port 4**: VC 2, VC 4

**Connections**:
- Instrument 1 connects to VC 6, SpFi I/F, VC 4
- Instrument 2 connects to VC 6, SpFi I/F, VC 2
- Control Processor connects to VC 6, SpFi I/F
- Mass Memory Unit connects to VC 6, SpFi I/F

**Nodes**:
- **VC 6**
- **VC 4**
- **VC 2**
- **VC**

**Notations**:
- SpFi I/F: SpaceFibre Interface
- VC: Virtual Channel
SpaceFibre Virtual Point to Point Link

SpaceFibre Routing Switch

Instrument 1
- VC 6
- VC
- SpFi I/F

Instrument 2
- VC 6
- VC
- SpFi I/F

Control Processor
- VC 6
- VC

Mass Memory Unit
- VC 2
- VC 4

SpFi Port 1
- VC 6
- VC
- VC 6
- VC
- SpFi I/F

SpFi Port 2
- VC 6
- VC
- VC
- SpFi I/F

SpFi Port 3
- VC 6
- VC

SpFi Port 4
- VC 6
- VC
- VC 2
- VC 4
- SpFi I/F
Spacecraft Data Handling Application

[Diagram of Spacecraft Data Handling Application with labels for VC, SpFi, SpFi Port, SpaceWire Routing Switch, Control Processor, Mass Memory Unit, Downlink Telemetry, Instruments, SpaceWire - SpaceFibre Bridge, and SpFi I/F.]
Spacecraft Data Handling Application

Instrument 1

Instrument 2

SpaceFibre Routing Switch

Control Processor

Mass Memory Unit

Downlink Telemetry

SpaceWire SpaceFibre Bridge

Instruments 1 3 4 5 6

Instruments 1 3 4 5 6
Impact on spacecraft mass

- **Mass saving**
  - 13 cables 2m each 90g = 2.34 kg

- **Replaced with**
  - 6 cables 2m each 60g = 0.72 kg
  - Saving of 1.6 kg

- **Plus redundancy**
  - Saving of 3.2 kg
Addition FDIR capability

Not allowed to use this Port since no VC 4

SpaceWire packet discarded

Can only route to same VC
VN routing to all destinations
VN routing to restricted destinations

Not allowed to route to this port
SpaceFibre Network Level

- Overview of intended SpaceFibre network operation
- Targeted at spacecraft data-handling applications
- VC buffers configured to specific VCs
- Can only route information within same VC/VN
- Provides additional FDIR capability
- Using redundancy of information
  - (address and VC number)
  - to detect faults
- Scheduling can be used with VNs for deterministic control applications
SpaceFibre Standard Update
SpaceFibre Standard Update E1

- Retry layer completed.
- Lane initialisation state diagram simplified.
- Data word identification state diagram simplified.
- Control symbols changed to improve robustness.
- Service interfaces improved (not yet finished).
- Quality of service specification improved.
- Management parameters added (more work yet).
- Electrical connectors and cable assemblies added to physical layer.
- Other corrections and clarifications throughout the document.
Physical Layer Connectors

- SpaceFibre specification
  - Characteristics of cable and connectors
  - If possible open specification of specific implementation

- Based on high performance connectors
  - Axon AxoMach
  - Designed for spaceflight applications
  - Very high bandwidth
  - Cable assemblies
  - PCB mounting connectors
  - Cable mass 60 g/m for SpFi full duplex
  - Connector mass approx 25 g
axoMach™ (R&T CNES)

- 2 coaxials space cable media ax2.4S per way
- 100 Ohm differential impedance
- Up to 10Gb/s per way (1, 2 & 4 ways)
- Low skew
- Low size
- E.M.I. improved
- Surface mount & parallel gap PCB terminations

Patent
N°PCT/FR2007/051
144
Design: heritage of microD

- MicroD space contacts
- PTFE inserts (Zc=100 Ohms)
- twin-coaxials (2 times ax2.4S)
- EMI gasket
- Retractable jack screw

Female inline

Male inline
SpaceFibre Flight Cable Assembly

- Connector body
- Backshell
- Individual shields
- TX +
- TX -
- RX -
- RX +
- PCB tracks
- SpaceFibre codec
- Connector female
- Connector male
- Cable assembly
SpaceFibre Flight/EGSE Cable Assembly
SpaceFibre EGSE Cable Assemblies

- External Serial ATA (eSATA) connectors used
  - Low cost
  - Small
  - High performance

- Crossover cable assemblies
SpaceFibre EGSE Cable Assembly
SpaceFibre EGSE Cable Assembly
SpaceFibre Flight/EGSE Adaptor
Quality of service

- Reminder about the QoS mechanism
  - Bandwidth reserved
  - Priority
  - Scheduled
Medium Access Controller

Output VCBs

Medium Access Controller

Frames

FCTs

RX FCT DECODE
Medium Access Controller

- Determines
  - Output VCB permitted to send next frame
- Depends on:
  - Which output VCBs have data to send
  - Which input VCBs at other end of link have room
  - Arbitration or QoS policy in force for each virtual channel
QoS: Bandwidth Reserved

Precedence

Bandwidth Credit Counter

① ② ③

time
QoS: Bandwidth Reserved

Precedence

Time
QoS Priority

Priority 1

Priority 2

Priority 3

Time
FDIR Support in QoS

- Bandwidth credit counter also supports fault detection:
  - Excessive bandwidth utilisation
    - When BW credit counter reaches negative limit
  - Under utilisation of allocated bandwidth
    - When BW credit counter stays at maximum limit for long period of time
- Can be used to detect
  - Babbling idiots
  - Faulty units
- All provided with simple, low cost, mechanism
Scheduled Precedence

- Time divided into time-slots
  - E.g. 64 time-slots of say 1 ms each
- Each VC allocated time-slots in which it is permitted to send data frames
- During a time-slot
  - If allowed to send in that time-slot
  - VC competes with other VCs also allowed to send in that time-slot
    - Based on precedence (priority and BW credit)
- A fully deterministic system would have one VC allowed to send in each time-slot
### Scheduled Precedence

<table>
<thead>
<tr>
<th>Time-slot</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>VC 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>VC 2</td>
<td></td>
<td></td>
<td>8</td>
<td></td>
<td></td>
<td>8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>VC 3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>VC 4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>8</td>
</tr>
<tr>
<td>VC 5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>VC 6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>8</td>
</tr>
<tr>
<td>VC 7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VC 8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Time-slot</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
</tr>
<tr>
<td>-----------</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>VC 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VC 2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VC 3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VC 4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VC 5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VC 6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VC 7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VC 8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Mixed Deterministic and Priority/BW-Reserved

<table>
<thead>
<tr>
<th>Time-slot</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>VC 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VC 2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VC 3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VC 4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VC 5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VC 6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VC 7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VC 8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Overview

1. Retry layer completed
   - Fastest error recovery time (3-5us at 2.5Gb/s)
   - Supports any link speeds (no timeouts used)
   - Recovers from nominal errors and detects non nominal link error conditions (i.e. multiple bit flips)

2. Lane initialization state machine simplified
   - Same functionality, 40% reduction in complexity.

3. More robust control word encoding scheme
   - Any single bit flip in the line is ALWAYS detected by the combination of encoding and retry layer.

4. SpaceFibre specifications completed
   - Management and physical layer described
Retry Layer features

Recovers from an error as fast as possible.

– A NACK is sent when any type of error is detected.
  - 8B10B disparity error
  - CRC error
  - Out of sequence frame (data, idle, FCT..)

– Retry starts as soon as a NACK is received.
  - Current data frame is aborted

– Broadcasts have higher priority than data frames at all times. (i.e. even when an error occurs)
Retry Layer features (2)

Reliable link under any type of error

– Including the loss of NACKs, ACKs or any other word.
– Nominal errors (i.e. single bit flips) are always recovered.
– Non-nominal conditions (i.e. multi bit flips) are detected before there is a change that the link reliability is compromised.
– Fatal errors such as unexpected hardware resets do not affect the link operation and user data integrity.

Supports any link speeds without configuration

– It does not use timeouts, so it can work with any link speed (i.e. multiple lanes or SpaceWire links)
If we retry every time a NACK is received, a single error can produce more than one retry event.

If we send a single NACK for the first error, then this NACK could get lost.
Retry Layer actual operation

- There are positive and negative sequence numbers (sequence polarity)
  - The receiver changes the polarity when an error occurs.
  - The sender changes the polarity when a NACK is received

The worst case recovery time is the round trip delay plus the transmission time of a data frame.

- This is less than 4us with a 100m cable length @2.5Gb/s in a prototype implementation.
Retry Layer additional features

- Broadcasts that are delayed due to a retry event are marked with a flag in EBF word.

- When a retry occurs, new broadcasts have priority over data to be resend.

- A single ACK word can acknowledge more than one frame.

- ACK word usage of bandwidth is limited.
  - 16 words must be sent between ACK words.
  - This helps to guarantee the bandwidth allocated to data frames.
Retry Layer TX implementation

Three Separate retry buffers for:
1. Broadcast Frames
2. FCTs
3. Data frames

Retry buffer control module removes frames acknowledged by the ACKs received
The retry buffers hold frames and FCTs until they are acknowledged.

One or more retry buffer can get full if:

1. The retry buffer is small and can not hold the amount of data sent before an ACK is received.
2. One or more ACKs are lost due to the cable being disconnected and connected.

- We need to request to send a new ACK using the FULL word

- When the retry buffer is full, a FULL word is sent

- When a FULL word is received, and ACK is sent
Retry Layer RX implementation

The Demux is implemented with the word identification state machine, which keeps track of which frame type is currently receiving.

If the CRC is correct the sequence number is checked against the expected value.

A Data frame and Broadcast frame buffer is used to temporally store one frame until the CRC at the end is checked.
Lane initialization state machine

- State machine now described using only 11 states instead of the original 19 states.
  - Error recovery state machine is removed as its functionality is already covered by the retry layer.

- Lane is disconnected when the lane receive error counter overflows.
  - This indicates that the BER is too high.
  - BER is measured by decreasing the error counter every N words (i.e. leaky bucket algorithm).

- Functionality is the same as the previous version.
  - Supports AutoStart, inverted RX polarity, Standby and LossOfSignal or receive error overflow indication.

- Implementation requires less than half of the original hardware resources.
No Signal means no signal detected at receiver inputs.
Typical initialization sequence

1. **Disabled**
   - Leave state when lane enabled

2. **Wait**
   - Leave state when in Start mode or signal is received.

3. **Started**
   - Sends INIT1s to start connection
   - Leave state when INIT1 or INIT2 received

4. **Connecting**
   - Sends INIT2s to indicate that it is receiving words.
   - Leave state when INIT2 or INIT3 received

5. **Connected**
   - Sends INIT3s to exchange connection parameters
   - Leave state when INIT3 received

6. **Active**
   - Sends user data
   - Leave state when in either end an errors occurs or is disabled.
Link Initialisation - Handshaking

INIT1  INIT1  INIT1  INIT1
INIT1  INIT2  INIT2  INIT2
INIT2  INIT2  INIT2  INIT2
INIT2  INIT2  INIT2  INIT2
INIT3  INIT2  INIT2  INIT2
INIT3  INIT3  INIT3  INIT3
INIT3  INIT3  INIT3  INIT3
INIT3  IDLE   IDLE   IDLE
INIT3  IDLE   IDLE   IDLE
INIT3  IDLE   IDLE   IDLE
INIT3  IDLE   IDLE   IDLE
INIT3  FCT +1 (1)
INIT3  FCT +32 (0)
IDLE   FCT +33 (1)
IDLE   FCT +34 (2)
IDLE   FCT +35 (3)
IDLE   FCT +36 (4)
FCT +1 (1)  FCT +37 (5)

Started  Connecting  Connected  Active

INIT1  INIT1  INIT1  INIT1
INIT1  INIT2  INIT2  INIT2
INIT2  INIT2  INIT2  INIT2
INIT2  INIT2  INIT2  INIT2
INIT3  INIT2  INIT2  INIT2
INIT3  INIT3  INIT3  INIT3
INIT3  INIT3  INIT3  INIT3
INIT3  IDLE   IDLE   IDLE
INIT3  IDLE   IDLE   IDLE
INIT3  IDLE   IDLE   IDLE
INIT3  IDLE   IDLE   IDLE
INIT3  FCT +1 (1)
INIT3  FCT +32 (0)
IDLE   FCT +33 (1)
IDLE   FCT +34 (2)
IDLE   FCT +35 (3)
IDLE   FCT +36 (4)
FCT +1 (1)  FCT +37 (5)
Robust control words encoding

- A control word begins with a comma
  - Comma symbol K28.7 is used so that a single bit flip error in a data word cannot convert it into a control word.

- Second character indicates type of control word.
  - They are selected to have polarity so any single bit flip occurred before will produce a 8B10B disparity error.

  - They are selected to maximize the hamming distance (number of different bits) between them, so more than three bit flips are required to convert one valid character into another valid one. The hamming distance is 4.
Medium Access Control and QoS rules

- A Virtual Channel can send a data frame:
  - If there is user data in the input buffer
  - If there is space at the destination
  - If it is scheduled to send data in the current TimeSlot.

- The Virtual Channel (VC) with higher precedence will be selected.
  \[ \text{Precedence} = \text{Priority Precedence} + \text{BW credit} \]
  - Priority precedence depends on the priority level of the VC.
  - BW credit depends on the amount of data sent respect to the reserved bandwidth of the VC.
  - BW credit only matters on VCs with the same priority

- Indicate to the user if a VC is using too much bandwidth or it is idle more than expected.
QoS schemes supported

- **Deterministic** scheduling scheme
  - All VCs scheduled at different timeslots.

- Simple **priority** scheme
  - Each VC is set to a different priority level.

- Simple **Bandwidth allocation**
  - All VCs set to the same priority level.

- Different priorities with bandwidth allocation
  - Bandwidth arbitration between multiple VCs for payload data
  - Command and Control set to use higher priority VCs.

- Different priorities with bandwidth allocation and scheduling
  - Some slots are reserved for VCs with deterministic requirements.
## SpaceFibre versus Rapid IO

<table>
<thead>
<tr>
<th>Feature</th>
<th>Rapid IO (Xilinx srio_ds696)</th>
<th>SpaceFibre</th>
</tr>
</thead>
<tbody>
<tr>
<td>Automatic retry</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Power management</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Scrambling</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Worst case retry reaction time</td>
<td>Timeout value</td>
<td>Delay of the line</td>
</tr>
<tr>
<td>Virtual Channels</td>
<td>1</td>
<td>8</td>
</tr>
<tr>
<td>Flow control</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Priorities</td>
<td>Yes (3 levels)</td>
<td>Yes (8 levels)</td>
</tr>
<tr>
<td>BW allocation</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Scheduling</td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>
SpaceFibre smaller than RAPID IO

Rapid IO v2.1 x1 (Based on Xilinx srio_ds696, Spartan 6 results)
SpaceFibre cost of multiple VCs

### Complexity: 1 VC vs 8 VC

<table>
<thead>
<tr>
<th></th>
<th>LUTs</th>
<th>FF</th>
</tr>
</thead>
<tbody>
<tr>
<td>SpaceFibre CODEC (1 VC)</td>
<td>2530</td>
<td>1825</td>
</tr>
<tr>
<td>SpaceFibre CODEC (8 VC)</td>
<td>4850</td>
<td>3470</td>
</tr>
</tbody>
</table>
SpaceFibre Next Steps
SpaceFibre FDIR

- **FDIR**
  - Fault detection
    - Parity/disparity
    - Invalid 8B/10B codes
    - Enhance Hamming distance
    - CRC
    - Over and under utilisation of expected bandwidth
  - Fault isolation
    - Galvanic isolation
    - Data framing – time containment
    - Virtual channels – bandwidth containment
  - Fault recovery
    - Link level retry
    - Graceful degradation on lane failure
    - Babbling idiot protection
    - Error reporting
SpaceFibre Next Steps

- Provide TKL2711 interface to CODEC
- Complete Multi-Lane layer specification
- Include Multi-Lane layer in CODEC
- Complete SpaceFibre specification
- Add Network layer