SpaceWire-D

Steve Parkes, Albert Ferrer
University of Dundee

Stuart Mills, Chris McClements, Alex Mason,
Paul McKechnie, Pete Scott, Bruce Yu,
STAR-Dundee Ltd
New SpaceWire Protocols

- **SpaceWire Protocol ID**
  - ECSS-E-ST-50-51C
  - Identifies packets as belonging to a particular protocol

- **Remote Memory Access Protocol (RMAP)**
  - ECSS-E-ST-50-52C
  - Read from and write to memory in a remote node over a SpaceWire network
  - Ideal for configuration, control, housekeeping and data collection
  - Already being used on several missions

- **CCSDS Packet Transfer Protocol**
  - ECSS-E-ST-50-53C
  - Transfers CCSDS Space Packets over SpaceWire
Deterministic SpaceWire (SpW-D)

- SpaceWire for control applications
- Determinism is essential
  - Determinism means
    - Predictable
    - Delivery within time constraints
  - Constrained Architecture
  - Time-slicing

- Single SpaceWire link is deterministic
Determinism with Constrained Architecture

Controlling Node

Data-Handling Processor

SpaceWire Router

Inst. 1
Inst. 2
Inst. 3
Inst. 4
Inst. 5
Mass Memory

Read Command
Read Reply
Read Command
Read Reply
Problem with Multiple Masters

Data-Handling Processor

Read Command

SpaceWire Router

Inst. 1
Inst. 2
Inst. 3

SpaceWire Router

Inst. 4
Inst. 5

Mass Memory
Determinism with Time-Slots

- Time-codes used to define time-slots
- Time-slot has same number as time-code that starts the time-slot
- 64 time-slots
Determinism with Time-Slots

Each initiator has a schedule table

Data-Handling Processor

<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>...</td>
</tr>
</tbody>
</table>

SpaceWire Router

Inst. 1

Inst. 2

Inst. 3

Mass Memory

<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>...</td>
</tr>
</tbody>
</table>

Specifies in which time-slots an initiator is allowed to initiate an RMAP command
Time-Slot 2

Data-Handling Processor

SpaceWire Router

Inst. 1  Inst. 2  Inst. 3

Mass Memory

0 1 2 3 4 ...
Y N N Y N ...

0 1 2 3 4 ...
N Y Y N N ...

Read  Command  Read  Reply
Time-Slot 3

Data-Handling Processor

SpaceWire Router

Inst. 1  Inst. 2  Inst. 3

Mass Memory

Read Reply

0 1 2 3 4 ...
Y N N Y N ...

0 1 2 3 4 ...
N Y Y N N ...

11
SpW-D Performance
Initiator Constraints

- Max data in RMAP read or write is limited
- Must respond to time-code quickly
  - Time-code to send RMAP command $< 5 \mu s$ (a)
- Must handle reply in a timely fashion
Target Constraints

- No modifications to RMAP target
- Must respond to RMAP command quickly
  - End of header to authorisation: < 5 μs (d)
  - Read or Write at least as fast as SpaceWire link can handle data 20 Mbytes/s
  - Read or Write latency: < 5 μs (f)
  - Create reply: < 5 μs (g)
- Can simply state that
  - Target must respond to an RMAP command within
  - 15 μs + time to transfer the data (at full SpW link speed)
SpW-D Performance

Effect of Data Length on Time-Slot Interval and Average Data Rate

Assumes average packet size is \((4+L)/2\) bytes where \(L = \text{max length}\)

Time-Slot Interval (us)

Average Data Rate (Mbits/s)

Max Data Length

- Time-Slot Interval
- Average Data Rate
Concurrent Data Transfer

- **Data-Handling Processor**
- **Spacewire Router**
- **Inst. 1**, **Inst. 2**, **Inst. 3**
- **Mass Memory**

**Write Command**
- Write Reply

**Read Command**
- Read Reply

Doubles the effective network bandwidth

Time and Space Partitioning of network
- Time: using time-slots
- Space: using different links
Multi-Slot Data Transfer

- For large data transfers
- Allow RMAP transaction to run over many slots

<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>
</tr>
</thead>
<tbody>
<tr>
<td>Processor</td>
<td>45</td>
<td>89</td>
<td>87</td>
<td>48</td>
<td>45</td>
<td>96</td>
<td>87</td>
</tr>
<tr>
<td>Mass Memory</td>
<td>89</td>
<td>63</td>
<td></td>
<td>48</td>
<td>87</td>
<td>96</td>
<td></td>
</tr>
</tbody>
</table>
Multi-Slot Data Transfer: Time-Slot 1

Data-Handling Processor

Inst. 1
Inst. 2
Inst. 3

Mass Memory

Write Reply

Read Reply

SpaceWire Router

0 1 2 3 4 ...
Y Y Y Y N ...

0 1 2 3 4 ...
M M M N N ...

0 1 2 3 4 ...
M M M M N N ...

Write Command
Multi-Slot Data Transfer: Time-Slot 2

Data-Handling Processor

Write Reply

Spacewire Router

Read Reply

Inst. 1

Inst. 2

Inst. 3

Mass Memory

0 1 2 3 4 ...

Y Y Y Y N ...

0 1 2 3 4 ...

M M M N N ...

20
SpaceWire-D

- Built on SpaceWire and RMAP standards
- Uses time-codes to produce time-slots
- Schedules communication in time-slots
- Uses RMAP transactions
- Can support FDIR
- Simple constraints:
  - RMAP target
    - Speed of response to RMAP command
  - RMAP initiator
    - Speed of response to time-code
    - Limit to size of RMAP data field
- Very simple to implement
Discussion

- *It is easy to make things complicated and difficult to keep them simple 😊*
- *The simpler something is the easier it is to check that it works properly 😊😊*
Discussion

- Key Principles
  - Simplicity
  - RMAP target unchanged
    - Provided it meets some (reasonable) performance criteria
  - FDIR more important than throughput
Discussion

- Is everyone happy with
  - Simple, concurrent and multi-slot scheduling?

- Multiple transactions in a single time-slot?

- Who specifies time-slot duration, data size, etc
  - The standard or the system engineer?

- Segmentation
  - Responsibility of application or standard?
  - Only modify the initiator?

- Retry and redundancy
  - Should this be included in the protocol stack?

- FDIR
  - How important is this?
SpW-D Protocol Stack

- User Application
- PTP
- SpaceWire-R (Retry/Redundancy)
- RMAP
- SpaceWire-D (Scheduling)
- SpaceWire
- PnP
Discussion

- Is everyone happy with
  - Simple, concurrent and multi-slot scheduling?
Discussion

- Multiple transactions in a single time-slot?
Discussion

- Who specifies time-slot duration, data size, etc
  - The standard or the system engineer?
Discussion

- **Segmentation**
  - Responsibility of application or standard?
  - Only modify the initiator?
Discussion

- Retry and redundancy
  - Should this be included in the protocol stack?
Discussion

- FDIR
  - How important is FDIR?
Things to be considered

- Amount of data in RMAP transaction?
  - Currently 512 bytes
Things to be considered

- Duration of time-slot?
  - Currently 50 usec
  - Request for epoch of 1/64 second
    - Makes a time-slot 15.6 ms
    - Not very timely but may be adequated
Things to be considered

- Allow various specific time-slot durations?
- Allow system engineer to determine time-slot duration?
Things to be considered

- Data rates on links
  - Normally should all be the same
  - Do we allow links of slower data rate to be supported
  - May be reducing the size of packet that can be transferred?
  - Or using multiple time-slots?
Things to be considered

- Key thing is what do we specify/constrain in the standard?
- We could leave it open and have devices specify key parameters e.g.
  - Speed of response of target device
  - Data rate supported
Things to be considered

- **Multiple transactions in one time-slot?**
  - Would add flexibility to SpW-D
  - But make FDIR much more difficult
  - Trade-off between flexibility/complexity vs FDIR
Things to be considered

- **Segmentation**
  - Do we include segmentation of large RMAP transactions in the SpW-D protocol?
  - i.e. SpW-D able to perform any required RMAP transaction.
Things to be considered

- Retry and Redundancy
  - Do we include this in the protocol stack?
  - Along with FDIR mechanisms?
Things to be considered