From: University of Dundee (Chris McClemments)

Arising from the presentations I feel I should clarify a few points which were made about the SpW-b core

Comments to Crisa Presentation

----

1) Number of clocks

This is dependent on the configuration. The minimum number of clocks required for any SpaceWire interface is 2 clocks. 1 Receive and 1 for the state/transmit part. Use of additional clocks for extra transmit speed and transmit clock generation options are optional and dependent on the users requirements.

2) Combinational logic generating clocks.

Combinational logic generating clocks is required for the receiver when decoding the input bit-stream, This is standard for all SpaceWire implementations.
The timing for the receiver requires some controlled placement of the receiver decode circuitry and this is standard for all spacewire cores. Some transmit clock configurations are require combinational logic to generate the clock.
This uses a standard gated glitch free multiplexer for clock switching which can be found in any system where this function is required.

3) Clock domains.

control/data crossing clock domains. All interfaces of this type in the CODEC are synchronised accordingly.

Comments to ESA Leon2 presentation

----

1) Non-standard data interface.

The transmit and receive interfaces are FIFO interfaces with simple FIFO interfaces. As the CODEC was developed as part of a SpaceWire router contract and became IP then no requirement was added for AMBA or other open core interfaces. In a way the rqeuirements were to reduce complexity so only the SpaceWire protocol was handled and the User had a simple FIFO interface to add their own interfaces. As you suggested in your conclusions the need for an AMBA interface was clear

2) Receive interface is master of receive buffer/FIFO

I feel this is a misunderstanding of how SpaceWire works. This is a requirement for all SpaceWire interfaces. Flow control is employed in SpaceWire where a flow control token is sent to indicate 8 data characters can be received. Once a flow control token is sent then eight data characters may be received and must be placed in a buffer/FIFO by the receiver so they are avaiable to the user.

3) Receive data interface is hard to implement

In fact the receive data interface is replicating some of the functions of a standard FIFO for efficient flow control operations. See attached slides.