CME 435

Streaming Packet Bus Interface Unit (SBIU)

Device Functional Specification, v1.3

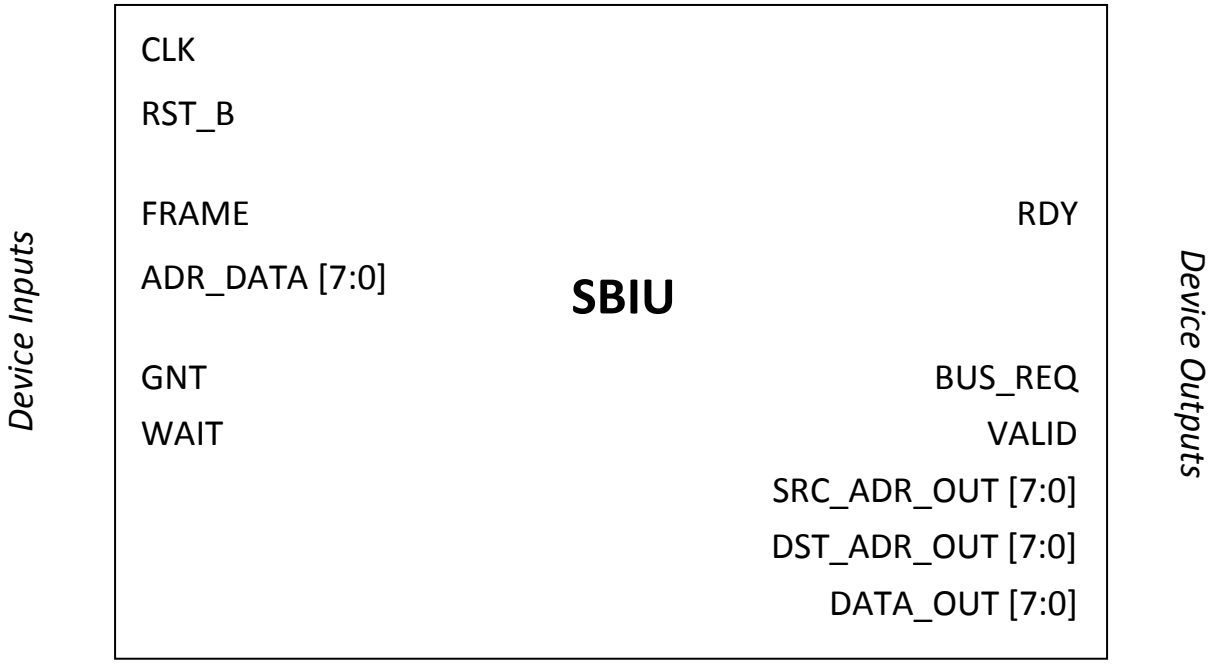
Prepared by: Anthony Rapley

1 Introduction

This document is the device functional specification for the Streaming Packet Bus Interface Unit (SBIU).

This device supports unidirectional unreliable packet traffic received by an external network interface and destined for an SoC core bus interconnect. The device provides end-to-end flow control, internal buffering, packet format translation, and checksum validation with invalid packet discard.

2 Block Diagram



3 Interfaces

3.1 Common

Name	Direction	Description
CLK	Input	Master clock. All interface packet transfer pins are synchronous to the rising edge of this clock.
RST_B	Input	Master reset. Active low. Asynchronous.

3.2 Upstream

Packets destined for the core bus interconnect are driven into the upstream interface of the SBIU.

Name	Direction	Description
RDY	Output	Upstream interface ready. Low while device is in reset or cannot accept new packets, high otherwise.
FRAME	Input	Packet framing indicator. Asserted high for the duration of a packet transmission, low otherwise. Consecutive packets must be separated by at least one clock cycle.
ADR_DATA [7:0]	Input	Multiplexed address/data bus.

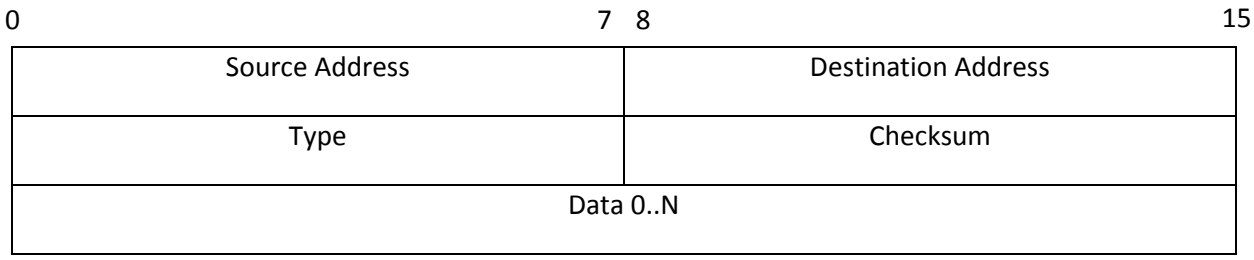
3.3 Downstream

Packets received by the upstream interface are forwarded to the core bus via the downstream interface of the SBIU.

Name	Direction	Description
BUS_REQ	Output	Bus arbitration request. Asserted high to request bus usage grant from the arbiter. Must remain high for the duration of packet transfer, low otherwise.
BUS_GNT	Input	Bus usage grant. Once asserted, remains high until BUS_REQ is deasserted.
WAIT	Input	Bus backpressure signal. The downstream interface must, on the next clock cycle, deassert its VALID signal and refrain from sending additional data packets until WAIT is deasserted.
VALID	Output	Indicates the presence of valid packet data on the DATA_OUT bus. Must be deasserted on the clock cycle subsequent to an assertion of the WAIT signal.
SRC_ADR_OUT [7:0]	Output	Packet source address output to core bus. Address remains asserted for the duration of the packet transmission.
DST_ADR_OUT [7:0]	Output	Packet destination address output to core bus. Address remains asserted for the duration of the packet transmission.

DATA_OUT [7:0]	Output	Packet data output to core bus.

4 Packet Format



Source Address – Identifies the sender of the packet. Ranges from 0 – 255.

Destination Address – Specifies the target of the packet. Ranges from 0-255.

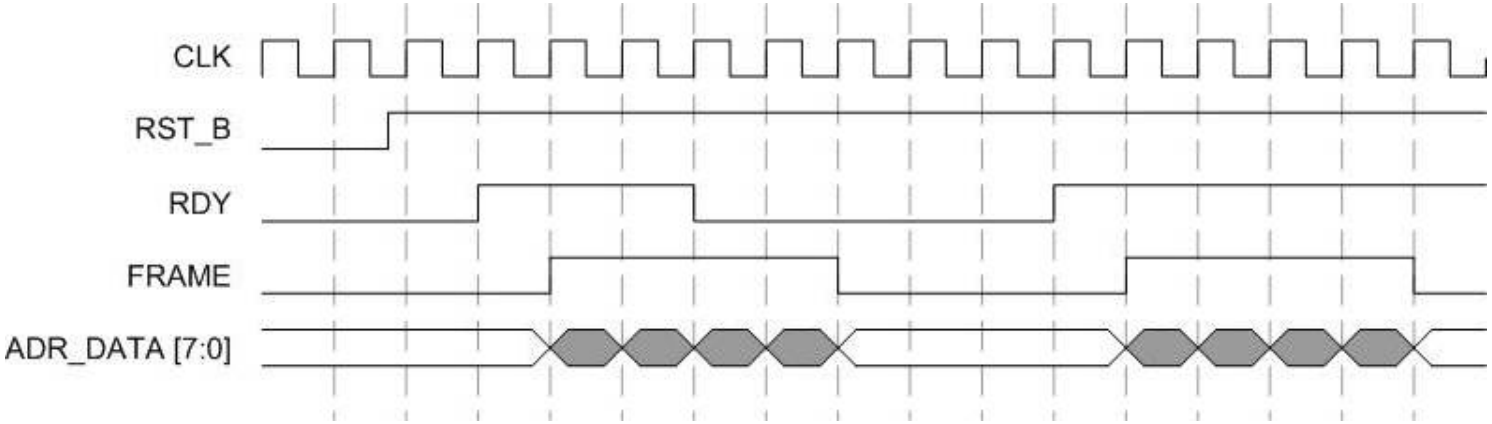
Type – Packet type identifier.

Value in Type Field	Mnemonic	Description
0	TX_DATA	Streaming data transmission. Data size may range from 0-28 bytes, inclusive.
1	CMD	Command packet. Data size statically 2 bytes.
2	HBEAT	Heartbeat “keep alive” message. Data size statically 0.

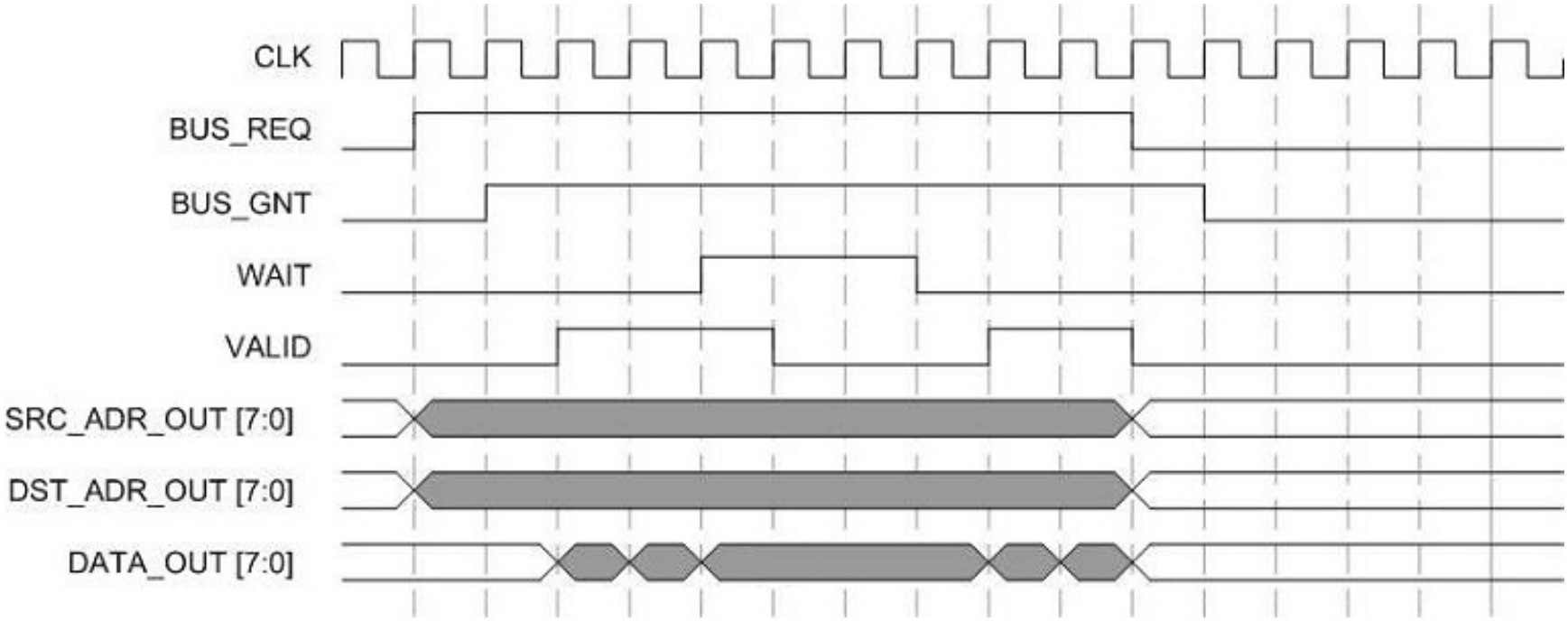
Checksum – Error check on all fields of the packet excepting the checksum field itself.

5 Timing

5.1 Upstream



5.2 Downstream



6 Operation

6.1 Reset

To prepare the device for operation, the device must detect a low-to-high transition on the active-low RST_B signal. The device shall synchronously assert its RDY signal on the upstream interface within 3 rising clock edges of the reset deassertion, indicating that it is ready to begin accepting upstream traffic and forwarding it to the core bus interconnect.

Upon assertion of the RST_B signal, all device outputs shall immediately transition low. The internal FIFO is additionally cleared.

6.2 Clock

The device operates on a single clock signal, CLK, which operates at a nominal frequency of 20MHz (50ns period).

6.3 Checksum

The checksum field is already set by an external device prior to being driven into the upstream interface of the SBIU. The checksum is validated within the SBIU before the packet appears on the downstream interface. If the checksum is invalid, the packet is silently discarded.

The checksum is calculated as the sum of each byte of the packet with overflow discarded, including the header but excluding the checksum field, which is then one's complemented. For example, if the packet bytes are 0x0D, 0x7A, and 0xB0, the sum with overflow discarded is 0x37 and the one's complemented result is 0xC8.

6.4 Packet Translation

6.4.1 Source/Destination Address Separation

When a packet received on the upstream interface is forwarded to the downstream interface, the source and destination address appear simultaneously on the SRC_ADR_OUT and DST_ADR_OUT buses, respectively.

6.4.2 Downstream Data Transfer

The packet data that appears on the DATA_OUT bus begins with the Type field, followed by the Checksum field, and any remaining data bytes. The Source Address and Destination Address do not appear on the DATA_OUT bus.

6.5 Internal FIFO

The device features an internal 64-byte FIFO, which is large enough to hold two maximum-sized (32-byte) packets.

6.6 Upstream Flow Control

When the internal FIFO no longer has room to accept a new maximum-sized packet (32 bytes), it shall deassert RDY. The device driving packets into the upstream interface is allowed to complete any current packet transfer, but may not begin transferring a new packet until RDY is reasserted.

6.7 Downstream Flow Control

6.7.1 Bus Arbitration

When the device has received a complete packet and validated the packet's checksum, it shall assert its BUS_REQ signal in combination with the packet source address on SRC_ADR_OUT and the destination address on DST_ADR_OUT. This combined assertion on the downstream interface shall occur within four clock cycles of the deassertion of FRAME on the upstream interface for the same packet. The device shall keep BUS_REQ, SRC_ADR_OUT, and DST_ADR_OUT asserted for the duration of the packet transfer. The external core bus arbiter will assert the BUS_GNT signal when the device has won arbitration. BUS_GNT will also remain asserted for the duration of the packet transfer. The device shall assert its VALID signal and commence data transfer on the rising clock edge following the assertion of BUS_GNT.

6.7.2 WAIT States

The recipient device on the other end of the core bus interconnect, upon seeing the destination address of the packet or after having received some data, may assert the WAIT signal into the device. When the WAIT signal is asserted, the device shall, on the next clock cycle, deassert its VALID signal and hold the value on DATA_OUT. When the WAIT signal is deasserted, the device shall, on the next clock cycle, reassert its VALID signal and begin placing new data on DATA_OUT.

In the event that the WAIT assertion coincides with the last byte of packet data, the device shall complete the packet normally. The WAIT signal is expected to be deasserted one cycle after the deassertion of the BUS_REQ signal.

Revision History

Version	Date	Changes
1.3	2010.10.28	Added better clarification and an example to checksum generation procedure in Section 6.3.