A Core Interface Architecture Supporting Multiple Network Packets per Processing Cycle

Prepared by:

William Goetzinger, Dr. Gary Delp, Dan Schraufnagel, Patrick Zabinski
Mayo Clinic
200 First Street SW
Rochester, MN 55905
goetzinger.william@mayo.edu
Phone: (507) 284-4056

November 23, 2011

Mayo-R-11-24-R0

Contents

List of Figures

1 Introduction

Future growth in network channel throughput requires network components to incorporate wider data buses.

As bus widths grow, the buses will become much wider than the smallest packets. Inefficiencies related to transporting small packets is a significant impediment to increasing network channel throughput.

This paper describes one interface that overcomes the challenges related to these trends. The interface allows more than one packet to be presented each clock cycle. For ease of discussion, the description revolves around the interface between an Ethernet Media Access Control (MAC) and the core logic internal to the network integrated circuit (IC). However, the interface can be applied throughout any system that benefits in supporting multiple packets per clock cycle.

2 Typical Interfaces

Figure 1 shows a typical present-day MAC transmit interface. The interface is used to transfer data within a network IC from the core logic to the MAC. A corresponding receive interface is required to transfer data from the MAC to the core logic.

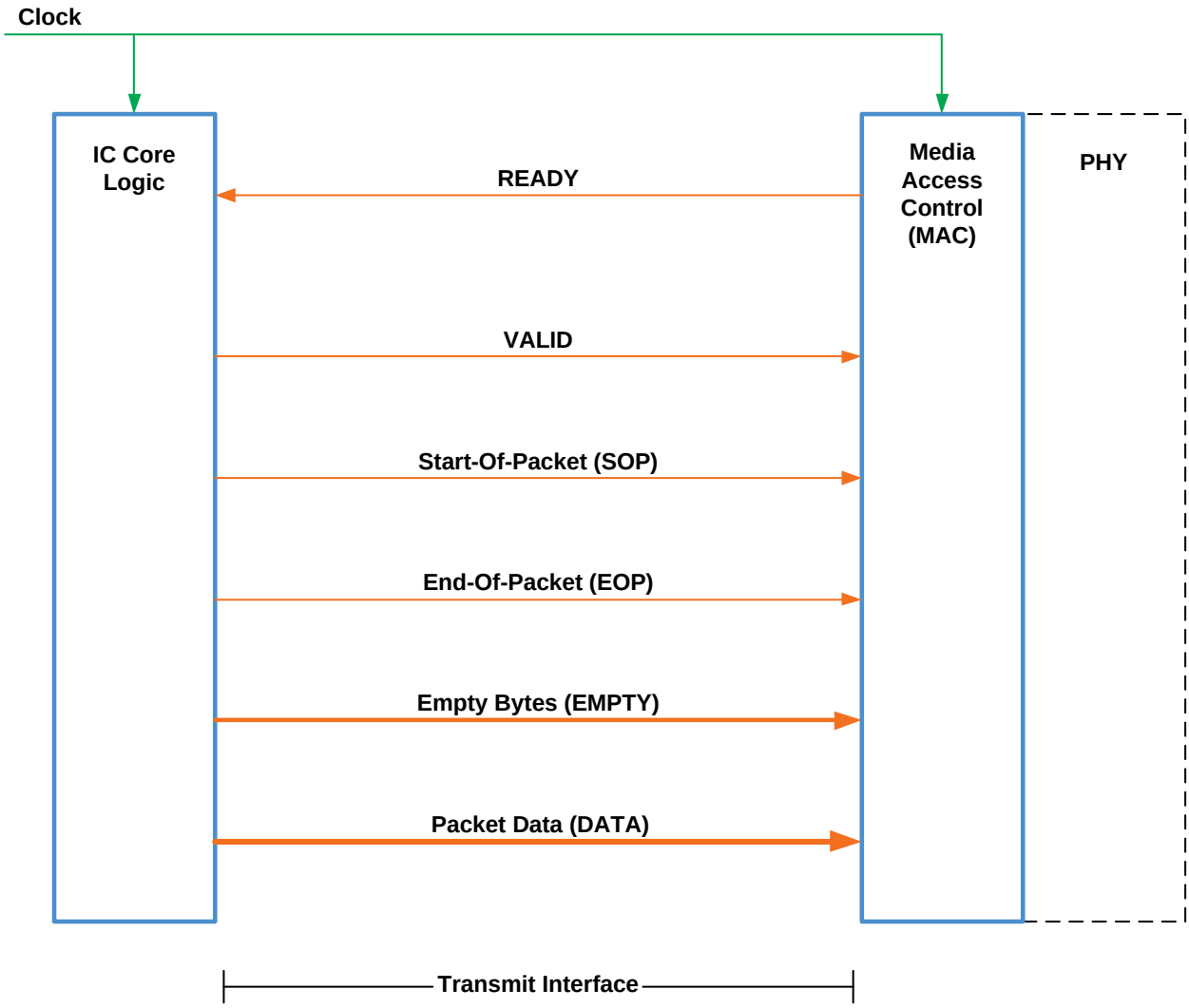

Figure 1: Typical MAC transmit interface. (42834)

The transmit and the receive interfaces both consist of the following signals:

- a ready indicator (READY)
- a valid indicator (VALID)
- a start-of-packet indicator (SOP)
- an end-of-packet indicator (EOP)
- an empty bytes value (EMPTY)
- the packet data (DATA)

If READY is asserted, the MAC is ready to accept new data from the core logic. If READY is not asserted, then the core logic must maintain the values on the signals to the MAC

until READY is asserted. The rest of the description in this section assumes that READY is asserted each clock cycle.

If VALID is asserted, then the SOP, EOP, EMPTY, and DATA fields from the core logic represent valid packet information. Otherwise, no valid packet information is present and these fields should be ignored. The rest of the description in this section assumes that VALID is asserted each clock cycle.

The DATA field contains the packet data. If SOP is asserted, then the DATA field contains the starting data of a new packet. If EOP is not asserted, then all of the DATA bytes contain packet data. If EOP is asserted, then the DATA field contains the ending data of a packet, and some of the DATA bytes might not contain packet data. The EMPTY value, which is only valid when EOP is asserted, indicates how many of bytes at the end of the DATA field do not contain packet data. These empty DATA bytes should be ignored.

For the typical MAC interface, the following restrictions apply each clock cycle:

- the DATA field may contain data associated with one and only one packet
- if SOP is asserted, then the data for the packet must be specified starting with byte 0 of the DATA field
- if the DATA field is wider than the minimum packet size, then SOP and EOP may be asserted at the same time, but only for a packet that starts and ends with the same DATA transfer

Because the DATA field may only contain data associated with one packet, the interface has the following characteristics.

First, assuming that the minimum packet size is smaller than the DATA bus width, then making the DATA bus wider does not increase the data rate when minimum-size packets are transported (Figure 2).

Second, the DATA bus will be inefficient whenever a small amount of packet data needs to be transported. For example, for a packet that is one byte larger than the DATA bus width, only one byte of DATA will be transmitted on the second clock cycle (Figure 3).

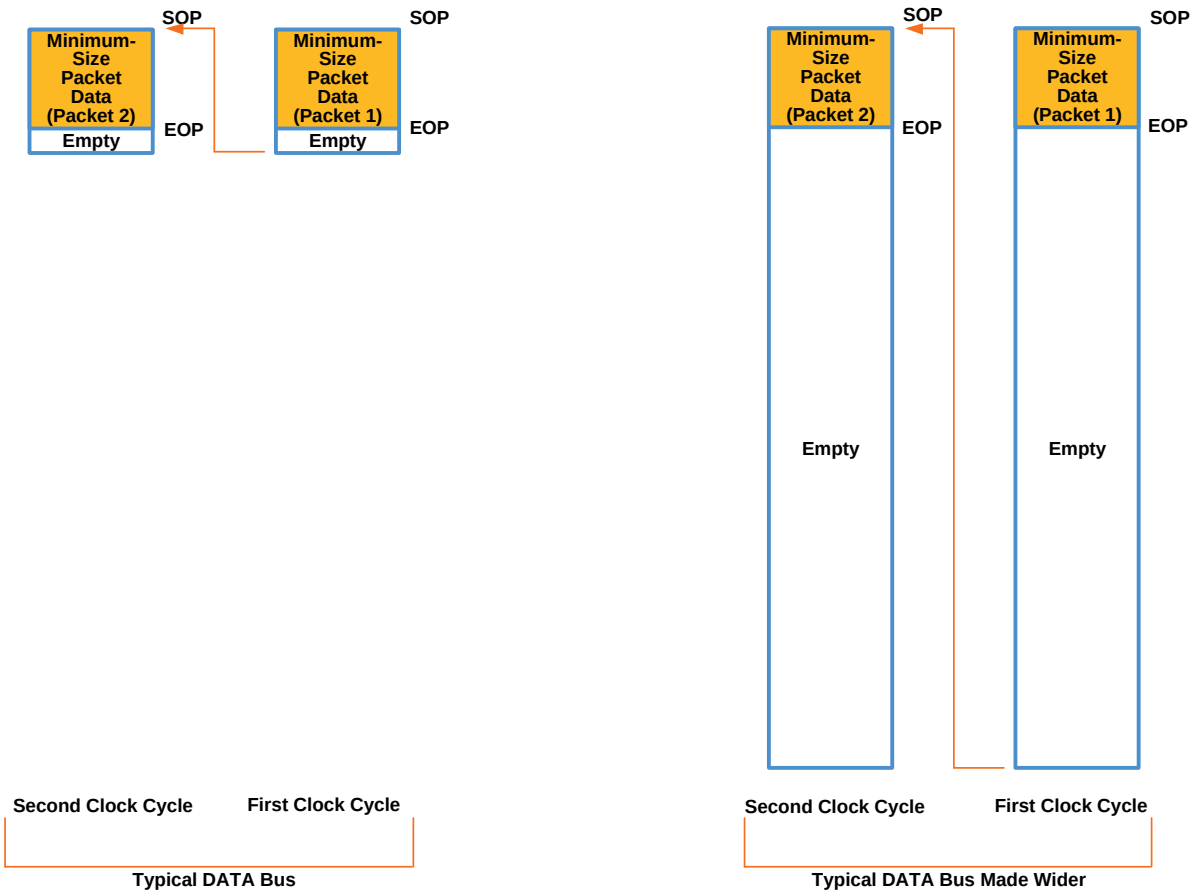

Figure 2: Minimum-size packets on typical DATA bus. (42835)

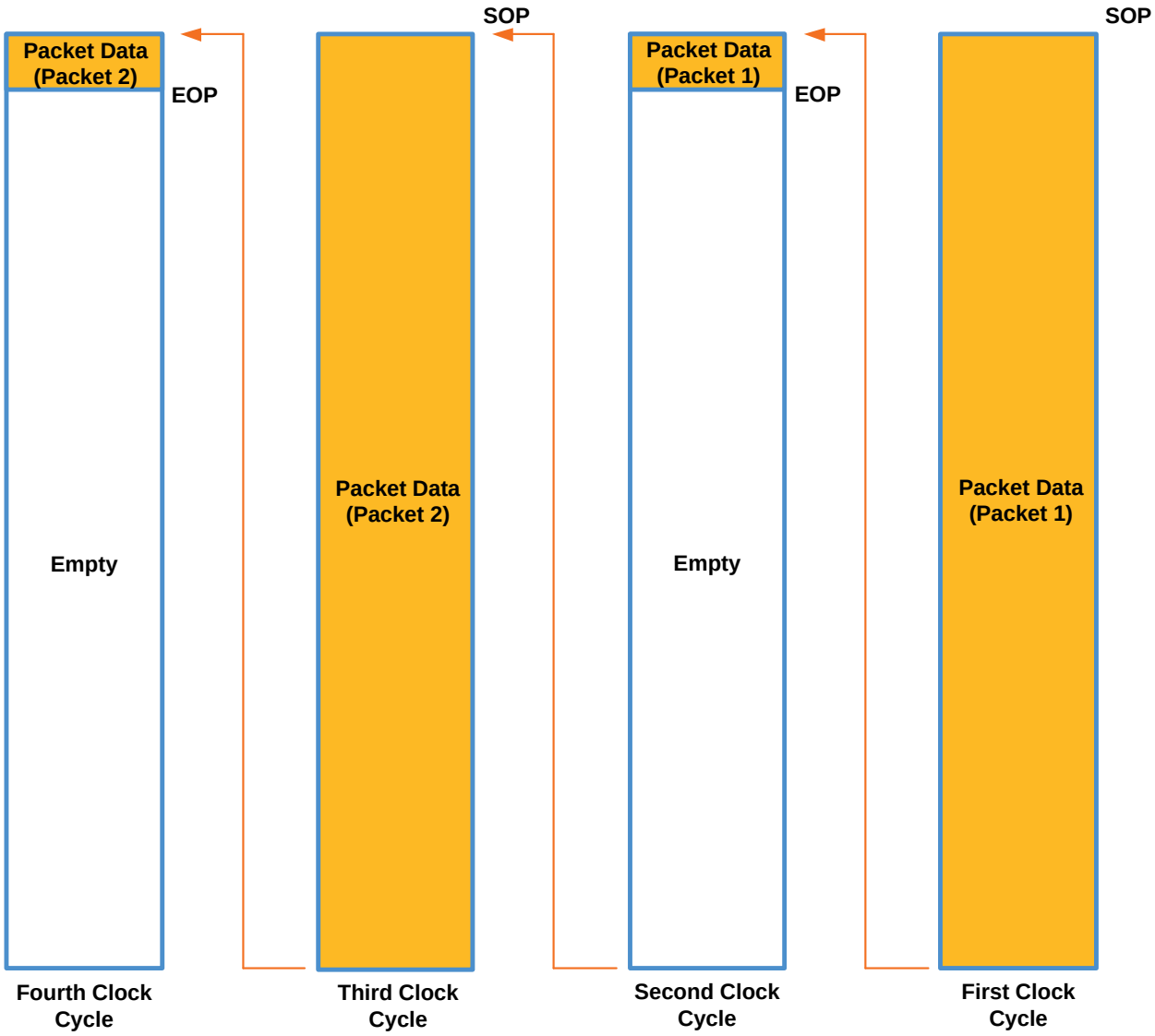

Figure 3: Packet transport efficiency on typical DATA bus. (42836)

3 Proposed Interface

Figure 4 shows a proposed transmit interface that supports multiple packets per clock cycle.

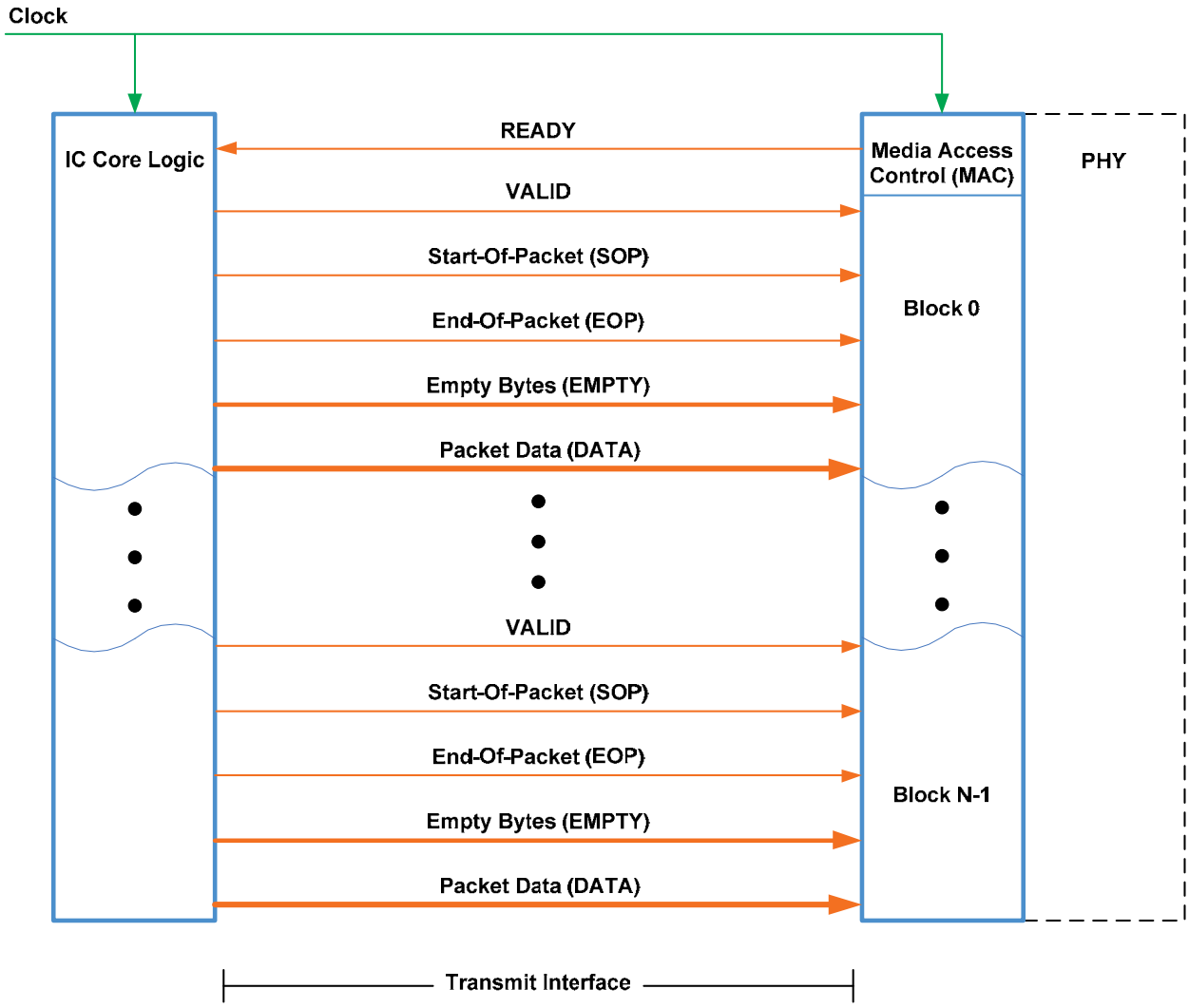

Figure 4: Proposed transmit interface. (42837)

This proposed interface is a concatenation of several interfaces of the type described in Section 2. Each of the interfaces is associated with a block of data and given a block number.

The maximum amount of data transmitted in a cycle is the number of blocks (N) multiplied by the number of DATA bytes in each block.

Each block has the following fields of signals associated with it:

- a valid indicator (VALID)
- a start-of-packet indicator (SOP)
- an end-of-packet indicator (EOP)
- an empty bytes value (EMPTY)
- a set of data (DATA)

In addition to the block-specific signals, an overarching READY signal is used for the overall bus. The descriptions of these fields and underlying signals are very similar to those of the previous section.

If READY is asserted, the MAC is ready to accept data from the core logic. If READY is not asserted, then the core logic must maintain the values on the signals to the MAC until READY is asserted. The rest of the description in this section assumes that READY is asserted each clock cycle.

If VALID for a block is asserted, then the SOP, EOP, EMPTY, and DATA fields for that block represent valid packet information for that block. Otherwise, no packet information is present for that block, and the remaining fields for that block should be ignored. The rest of the description in this section assumes that VALID is asserted.

The DATA field contains the packet data for the block. The size of the DATA field should be the same for all of the blocks. If SOP for a block is asserted, then the DATA field for that block contains the starting data of a new packet. If EOP for the block is not asserted, then all of the DATA bytes for the block contain packet data. If EOP is asserted, then the DATA field contains the ending data of a packet, and some of the DATA bytes might not contain packet data. The EMPTY value for a block, which is only valid when EOP for the block is asserted, indicates the number of bytes at the end of the DATA field for that block that do not contain packet data. These empty DATA bytes should be ignored.

This proposed interface differs from the interface of Section 2 in the following ways:

- the aggregate DATA field (i.e., the DATA fields for all of the blocks, taken together) may contain data for more than one packet
- if SOP for a block is asserted, then the packet data for that block must be specified starting with byte 0 of the DATA field for that block

Because the aggregate DATA field may contain data associated with more than one packet, the interface has the following characteristics.

First, making the aggregate DATA bus wider increases the data rate when minimum-size (or any size) packets are transported (**Error! Reference source not found.**).

Second, the DATA bus will be more efficient than the interface of Section 2 when a small amount of packet data needs to be transported. For example, for a packet that is one byte larger than the DATA bus width, only one byte of DATA will be transmitted in the first block of DATA on the second clock cycle. However, in that same clock cycle subsequent DATA blocks can be used to transmit data for other packets (Figure 6).

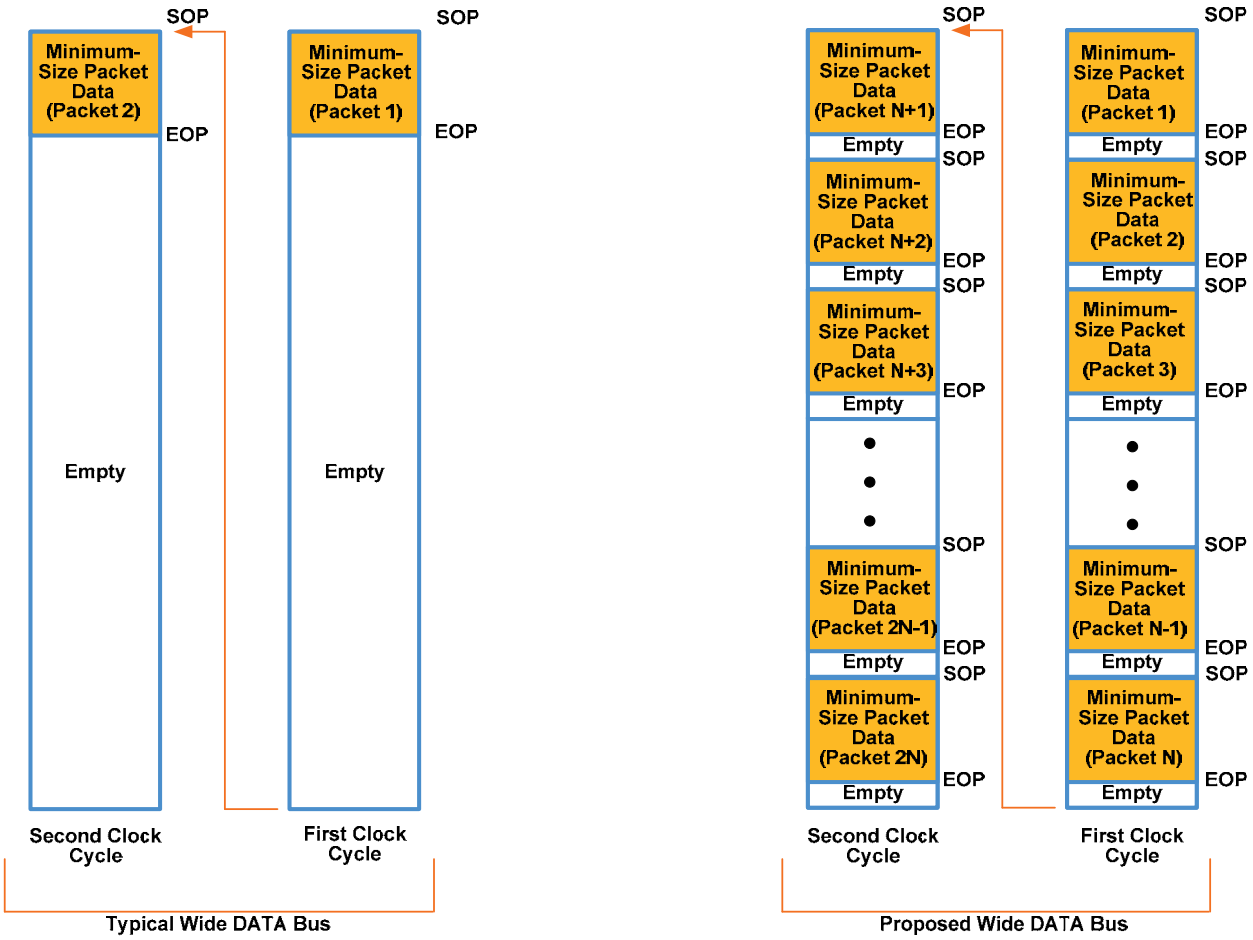

Figure 5: Minimum-size packets on proposed DATA bus. (42838)

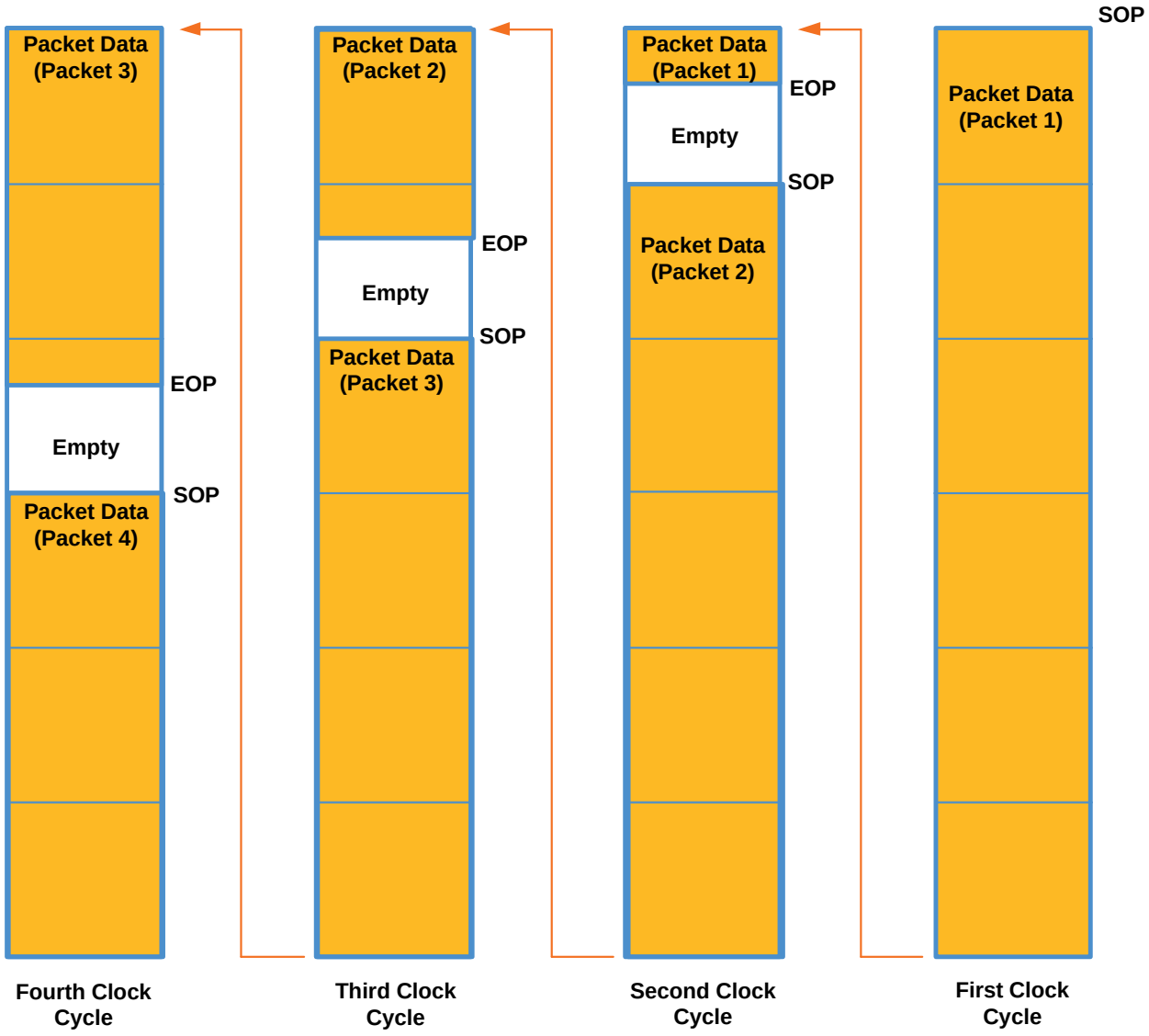

Figure 6: Packet transport efficiency on proposed DATA bus. (42839)

4 Variations of the Proposed Interface

The proposed interface is meant solely to provide a framework for the discussion of future wide data bus implementations. Possible changes to the proposed interface include, but are not limited to, the following:

- different aggregate DATA path widths
- different DATA block sizes (and accompanying changes to the size of EMPTY)
- modifications to accommodate certain packet characteristics (e.g., different packet size ranges)
- modifications to increase efficiency (e.g., eliminating the need for EMPTY by allowing SOP to point to any byte in a DATA block)
- modifications for various protocols

5 Summary

The continued growth in network channel bandwidth will require network components to incorporate wider data buses. A proposed implementation for a wide data bus interface was presented along with some possible variations.