# Memory Controllers Part 1 of 2

**By Michael Miller**
**Chief Technology Officer**
**MoSys, Inc.**

## Overview

With so many solutions out on the market today based on QDR SRAM, it makes sense to take a deeper dive into memory solutions and what they can offer. The MoSys memory controllers are designed to simplify the integration of the accelerator engines into a design to match standard SRAM interfaces. The controllers are built with a simple AXI like interface which hides away the external high-speed SerDes control and implementation of the GCI protocol essentially "hidden away" from your design effort. MoSys controllers which have been deployed in the field since 2004 have been proven to be robust and reliable.

The interface, which is presented to the user application interface is a straightforward Address, Data, Command bus structure, that is compatible with and easily adapted to an AXI interface. Multiple versions are available to support different access patterns and for different hosts (Xilinx, Intel, ASIC etc.)

The interface, which is presented to the user application interface is a straightforward Address, Data, Command bus structure, that is compatible with and easily adapted to an AXI interface. Multiple versions are available to support different access patterns and for different hosts (Xilinx, Intel, ASIC etc.)

The MoSys memory controllers are designed and offered with a few variations of memory access patterns. The most common access patterns are:

- Balanced Read/Write (similar to QDR SRAM)
- Native (higher read access than write – for table access applications)

- Burst (Allows one command to access 2, 4 or 8 locations)
- Statistics (Takes advantage of the Accelerator Engines on board ALU to keep data statistics)

The RTL that is supplied by MoSys provides an interface between the User Application Logic and the MoSys Accelerator Engine device (could be BE-2 or BE-3 families). The Memory Controller also implements all the required signaling and handshaking defined in the GCI Interface protocol, Framer logic.

The signals interface at the User Application provides Bandwidth Engine User a simple SRAM memory read/write operation with burst capability. This simple interface shields the users from the BE-X commands and the scheduling logic for Bandwidth Engine memory partitions wheel.

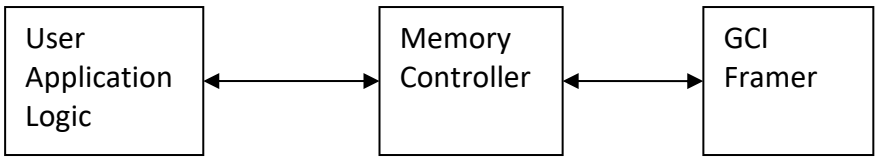


**Figure 1: Memory Controller Interface**

The goal of each of the Memory Controller designs is to balance the bandwidth between the User Application Interface and the Bandwidth Engine Interface. For many applications there will be four read/write interfaces from the User Application running at the host core clock frequency. This is to balance the bandwidth of the application logic (assumed to be running at FPGA speeds vs. the GCI I/O and core frequency of the MoSys accelerator engines). In many of the FPGA applications that has been between 250MHz and 390MHz clock rate. Each interface can accommodate one memory read and one memory write on each clock cycle. These result in memory accesses per interface that can saturate the access bus to the memory.

This allows the total bandwidth at the User Application interfaces to be up to 2 billion memory accesses per second when using a BE-2 device and 6 billion memory operations when using a BE-3 device. (This bandwidth matches the total I/O bandwidth on Bandwidth Engine when using all 16 lanes at maximum allowable SerDes rate of the Accelerator engine device) per lane. The following picture illustrates the above memory bandwidth discussion.

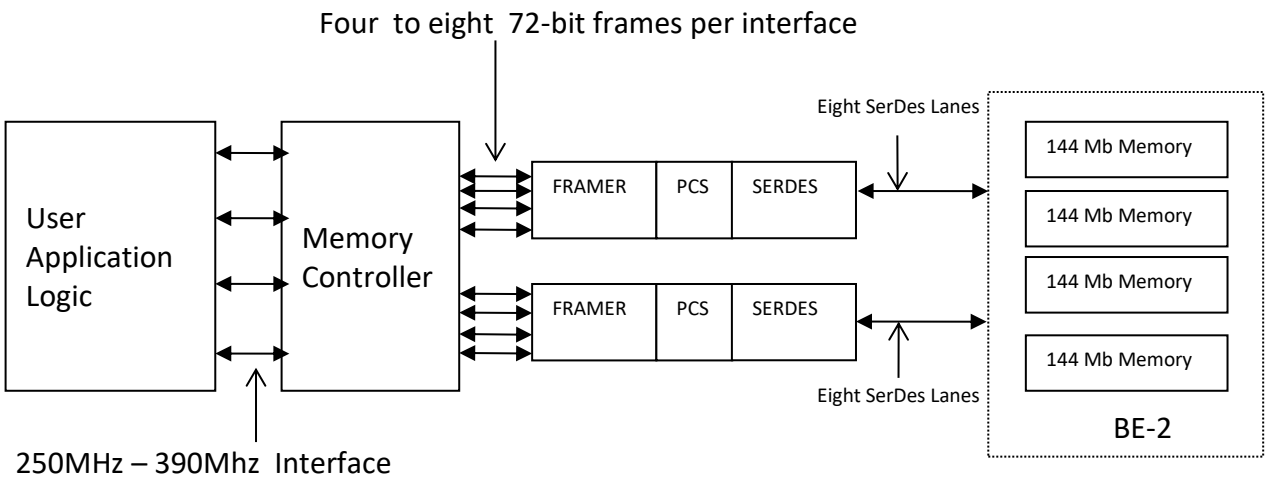


**Figure 2: Memory Bandwidth between Memory Controller and Bandwidth Engine.**

## Additional Resources:

BLAZAR Family Overview

RLT B2-B3 Memory Controller Product Brief

RTL Controller Overview

Part 1 of this blog focused on an overview of MoSys memory controllers. Part 2 will delve into the micro architecture of MoSys memory controllers and statistics of R-M-W. If you are looking for more technical information or need to discuss your technical challenges with an expert, we are happy to help. Email us and we will arrange to have one of our technical specialists speak with you. You can also sign up for updates. Finally, please follow us on social media so we can keep in touch.

# Memory Controllers Part 2 of 2

**By Michael Miller**
**Chief Technology Officer**
**MoSys, Inc.**

## Statistics (R-M-W)

In part 1 of this blog, we discussed memory controllers in a broad context. Part 2 will delve into the micro architecture of MoSys memory controllers and statistics of R-M-W

## MoSys BE Statistic Controller

The BE interface performs the same bridge function between the application logic and the bandwidth engine device however it is designed to support R-M-W or read-modify-write operations in order to allow the application to issue ALU operations to perform functions such as maintaining statistics or metering. This design utilizes the large memories found in the BE2 or BE3 devices along with its embedded ALU and its RMW instructions in order to implement the counter or metering.

As an example, with the current BE2 memory size and 8 serial lanes running at 12.8 Gbps, this Statistic Controller has the following capabilities:

- Incrementing up to eight concurrent counters at the rate up to 160 Million counts per second which is higher than typical 100GE at 148.8 Mpps.
- The 576 Mb memory capacity on BE2 can support up to 8 x 256K counters.
- Interface to read the counters. This operation returns the counter content.
- These counters are lifetime counters (which means they are a full 64 bits wide).
- Counters are initialized by the controller after de-assertion of reset.
- Diagnostic interface to read and write BE2 memory for initial memory testing.
- Optional ECC Correction circuitry to correct one-bit error and detect two bits or more error on the counter read data.

**Micro Architecture**

For a statistics and R-M-W operations, there are 8 counter interfaces in which four counters are mapped to lower part of the four partitions in BE memory and other four counters are mapped to upper part of the four partitions. The user specifies 18-bit counter index that is mapped to a location in partition. The figure 8 below illustrates the statistic controller interfaced with user at one end and GCI PCS/Framer at the other end.

The user requests are captured at the input of an Asynchronous FIFO at a clock frequency of the application logic. The R-M-W (or statistics) controller then pulls user requests out of this Asynchronous FIFO at its internal clock derived from SerDes clock. The statistic controller operates at this frequency to optimize the interface with GCI running at SerDes speed. The statistic controller creates RMW (Read-Modify-Write) commands from the user requests to implement counter operation using BE memory. The statistic controller has following major blocks:

1. Eight asynchronous FIFOs corresponding to 8 user increment requests.
2. Eight asynchronous FIFOs corresponding to 8 users read count requests.
3. One Asynchronous FIFO for memory Diagnostic interface user request.

4. A scheduler block which performs RMW command formation and scheduling based on user request.
5. A receive data block that receives the read data from the PCS/Framer block and send it to the user on its respective interface, this being the same interface that requested the action.
6. A debug interface block that tests the BE memory.

On each wheel (or FPGA Cycle time), four counter operations are performed that is four R-M-W commands corresponding to counter increments, are scheduled. On one-wheel cycle, four counters located at lower part of four partitions in BE memory, are incremented. In next wheel cycle, another four counters located at upper part of four partitions, are incremented. Out of every 15 accesses of the BE memory, first 14 accesses are RMW commands corresponding to counter increments and 15$^{th}$ access is the memory read operation. The RMW and read command scheduling is shown in Figure 1 below.

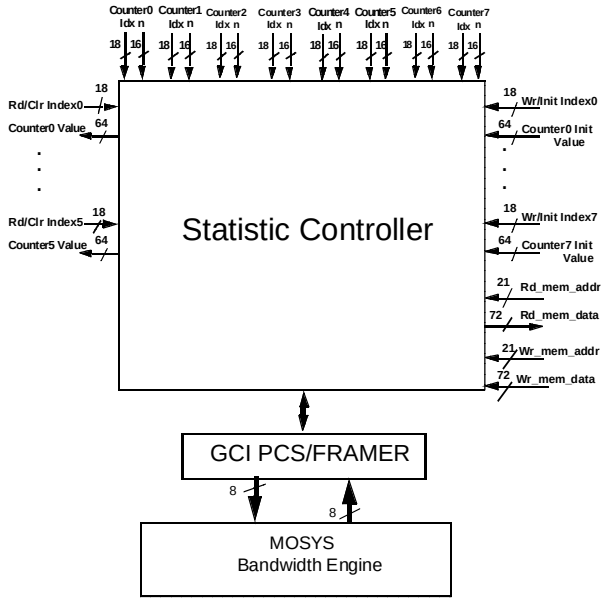


Figure 1: BE Statistic Controller.

**Summary**

MoSys memory controllers are designed to simplify the integration of the accelerator engines into a design. The controllers are built with all the high-speed SerDes control and implementation of the GCI protocol essentially "hidden away" from your design effort. MoSys controllers which have been deployed in the field since 2004 have been proven to be robust and reliable.

The interface which is presented to the user application interface is a straightforward Address, Data, Command bus structure, that is compatible with and easily adapted to an AXI interface. Multiple versions are available to support different access patterns and for different hosts (Xilinx, Intel, ASIC etc.)

This blog allows a user to realize that integration and implementation of the MoSys family of accelerator engines is not a long process and can be accelerated by utilization of the readily available MoSys IP.

**Additional Resources:**

Memory Controllers Blog Part I

BLAZAR Family Overview

RLT B2-B3 Memory Controller Product Brief

RTL Controller Overview

If you are looking for more technical information or need to discuss your technical challenges with an expert, we are happy to help. Email us and we will arrange to have one of our technical specialists speak with you.  You can also sign up for updates. Finally, please follow us on social media so we can keep in touch.

###