SPARC MBus-to-Futurebus+ Bridge Using the Texas Instruments Futurebus+ Chipset

Robert Gugel
Mixed Signal Product Group

SCAA019A
March 1996

TEXAS INSTRUMENTS

IMPORTANT NOTICE

Texas Instruments (TI) reserves the right to make changes to its products or to discontinue any semiconductor product or service without notice, and advises its customers to obtain the latest version of relevant information to verify, before placing orders, that the information being relied on is current.

TI warrants performance of its semiconductor products and related software to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are utilized to the extent TI deems necessary to support this warranty. Specific testing of all parameters of each device is not necessarily performed, except those mandated by government requirements.

Certain applications using semiconductor products may involve potential risks of death, personal injury, or severe property or environmental damage ("Critical Applications").

TI SEMICONDUCTOR PRODUCTS ARE NOT DESIGNED, INTENDED, AUTHORIZED, OR WARRANTED TO BE SUITABLE FOR USE IN LIFE-SUPPORT APPLICATIONS, DEVICES OR SYSTEMS OR OTHER CRITICAL APPLICATIONS.

Inclusion of TI products in such applications is understood to be fully at the risk of the customer. Use of TI products in such applications requires the written approval of an appropriate TI officer. Questions concerning potential risk applications should be directed to TI through a local SC sales office.

In order to minimize risks associated with the customer's applications, adequate design and operating safeguards should be provided by the customer to minimize inherent or procedural hazards.

TI assumes no liability for applications assistance, customer product design, software performance, or infringement of patents or services described herein. Nor does TI warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right of TI covering or relating to any combination, machine, or process in which such semiconductor products or services might be or are used.

Contents

Title *Page*

Introduction **3–197**

Transaction Support **3–197**

Bridge Architecture **3–199**

Command **3–200**

Address **3–200**
Direct-Memory Model 3–200
Page-Memory Model 3–200

Datapath **3–201**
Considerations 3–201
Byte-Lane Mapping 3–201

Control Logic **3–203**

Summary **3–206**

List of Illustrations

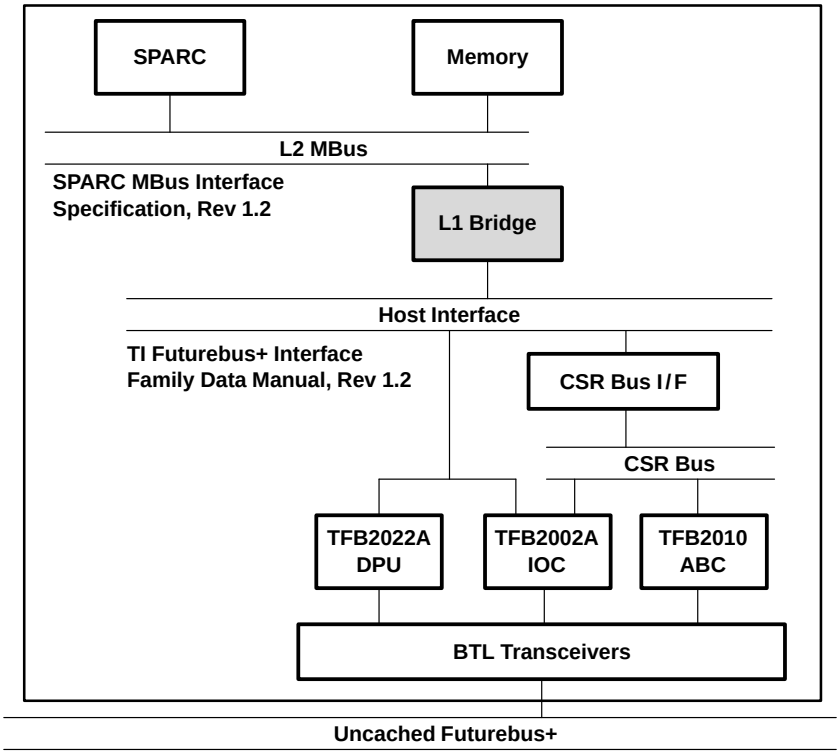
Figure *Title* *Page*

1 SPARC FB+ 3–197
2 MBus-to-HIF Bridge 3–199
3 MBus/FB+ Memory Map 3–201
4 Byte-Lane Mapping for a 64-Bit-Only System 3–202
5 Byte-Lane Mapping for a 64-/32-Bit System 3–202
6 Byte-Lane Mapping Showing Path Taken by Address Quadlet on HD 3–203
7 MBus-Word Read to HIF Single-Read Transaction 3–204
8 MBus-Word Write to HIF Single-Write Transaction 3–205

Introduction

This application report describes the logic necessary to connect the SPARC MBus to the Texas Instruments (TI) Futurebus+ (FB+) chipset host interface (HIF). This logic is a translator of MBus transactions to HIF transactions and vice versa. The MBus-to-HIF bridge described is MBus level-1 compliant. Level-1 transactions are the noncache-coherent subset of MBus level-2 transactions. Even though level-2 transactions are occurring on the MBus, they cannot cross the L1 bridge.

Figure 1 shows a block diagram of a FB+ module featuring a SPARC processor that can access FB+. The MBus-to-HIF bridge block is highlighted. Since knowledge of FB+, the HIF, and MBus is necessary to understand this definition, references to the applicable specifications also are included in Figure 1.



NOTE: Reference IEEE P896.1 and P896.2

Figure 1. SPARC FB+

Transaction Support

A transaction originating on the MBus must be transported to the HIF and then to the FB+. This requires the L1 bridge to be a slave to MBus transactions and a master to HIF transactions. Likewise, a transaction originating from FB+ is transported to HIF and then to the MBus. In this case, the L1 bridge is a slave to HIF transactions and a master of the MBus transactions. Each of the three buses has a set of transactions that must be mapped to each other when crossing bridges.

Table 1 shows the translation of MBus transactions to HIF transaction that is performed by the L1 bridge when it is a slave to the MBus and a master of the HIF..

Table 1. MBus-to-HIF Transaction Mapping

MBUS TRANSACTION TYPE	RESULTING HIF TRANSACTION TYPE
Byte read/write	DW32 single 1-byte read/write partial
Half-word (2 bytes) read/write	DW32 single 2-byte read/write partial
Word (4 bytes) read/write	DW32 single read/write
Double word (8 bytes) read/write	DW64 8-byte burst read/write
16-byte burst read/write	DW64 16-byte burst read/write
32-byte burst read/write	DW64 32-byte burst read/write
64-byte burst read/write	DW64 64-byte burst read/write
128-byte burst read/write	Two DW64 64-byte burst reads/writes chained together with HIF MORE signal

Table 2 shows the translation of HIF transactions to MBus transactions that is performed by the L1 bridge when it is a slave to the HIF and a master of the MBus. The data-width 32 (DW32) HIF bursts always cause 4-byte-wide transactions on MBus, and data-width 64 (DW64) HIF bursts always cause 8-byte-wide MBus transactions. This means that dynamic bus sizing from HIF to MBus is not supported by the architecture suggested.

Table 2. HIF-to-MBus Transaction Mapping

HIF TRANSACTION TYPE	RESULTING MBUS TRANSACTION TYPE
DW32 single 1-byte read/write partial	Byte read/write
DW32 single 2-byte read/write partial	Half-word (2 bytes) read/write
DW32 single 3-byte read/write partial	Half-word (2 bytes) read/write and a byte read/write
DW32 single read/write	Word (4 bytes) read/write
DW32 8-byte burst read/write	Two individual word (4 byte) reads/writes
DW32 16-byte burst read/write	Four individual word (4 byte) reads/writes
DW32 32-byte burst read/write	Eight individual word (4 byte) reads/writes
DW32 64-byte burst read/write	Sixteen individual word (4 byte) reads/writes
DW64 8-byte burst read/write	Double-word (8 bytes) read/write
DW64 16-byte burst read/write	16-byte burst read/write
DW64 32-byte burst read/write	32-byte burst read/write
DW64 64-byte burst read/write	64-byte burst read/write

Bridge Architecture

Figure 2 shows suggested implementation of the MBus to HIF bridge.

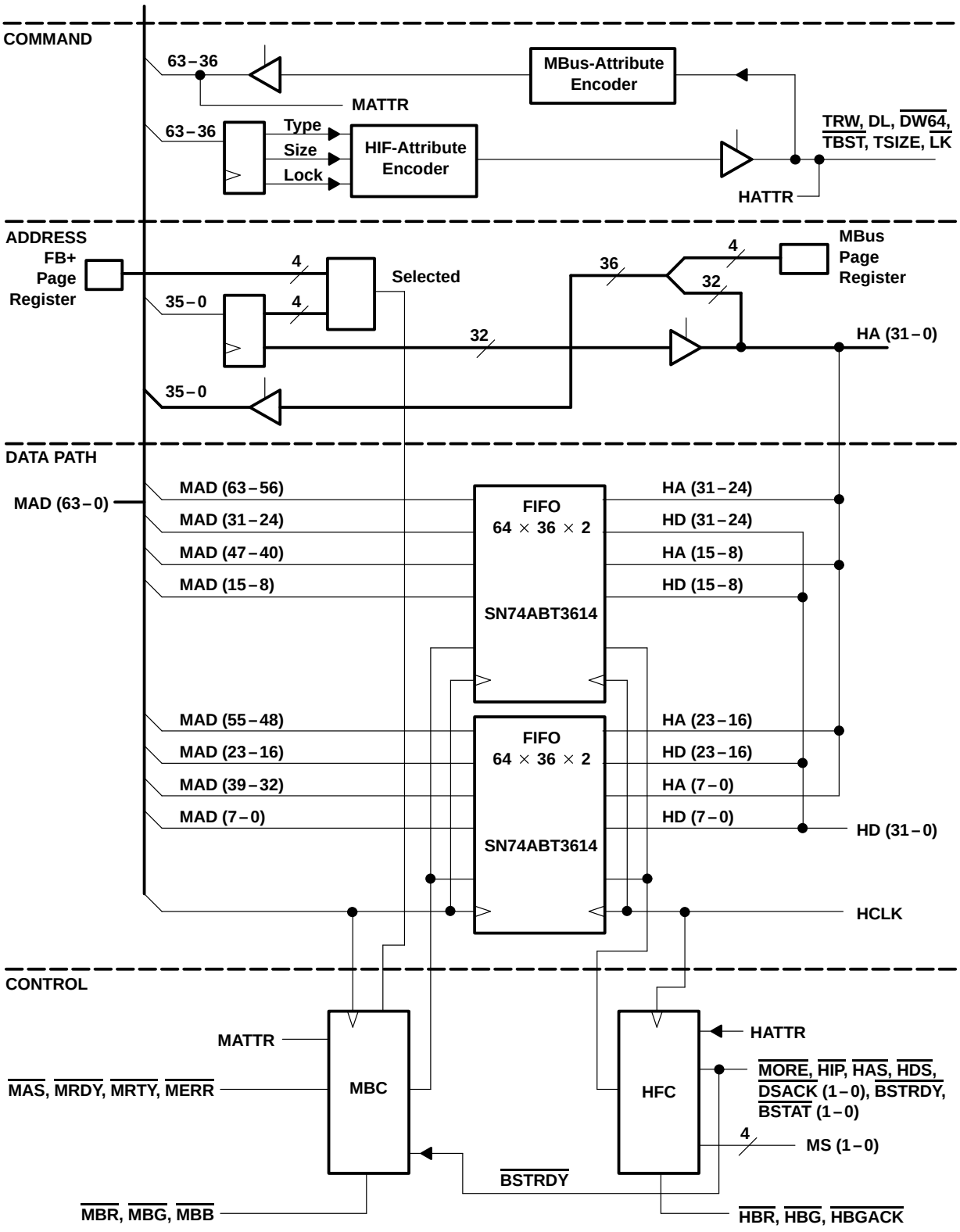


Figure 2. MBus-to-HIF Bridge

A typical bus-bridge implementation consists of command, address, datapath, and control logic. The operations of these sections in this design are as follows:

Command

The host interface defines a set of discrete signals that indicate the attributes (i.e., type, size, etc.) of the transaction taking place. The MBus does the same thing; however, these signals are multiplexed onto signals in the field MAD (63:0) that are not used to carry the address during the address phase of transactions. These command attributes also are logically encoded differently by the two buses.

When a slave to MBus, the command section of the L1 bridge must latch the transaction-specific information from MAD (63:36) during the address phase and encode it into host-interface attributes that correspond to the resulting transaction to be mastered on the host interface.

When a slave to the host interface, the command logic encodes the HIF's attribute signals into MBus attributes that correspond to the resulting transaction to be mastered on the MBus. These encoded MBus attributes must be multiplexed onto MAD (63:36) along with the MBus address during the MBus-address phase.

Address

The host interface has a 32-bit physical address space with a 36-bit extension. The MBus defines a 36-bit physical-address space. The address portion of the L1 bridge logic must do three things:

Recognize the address region that it must respond to as an MBus slave and as an HIF slave

Transport the physical address from one protocol to the other

Relate one bus-memory region to the other

Direct-Memory Model

The simplest memory model would be a logically direct connection between the MBus 36-bit address and the HIF 36-bit address. This would mean that all MBus addresses not within FB+ MEM_BASE and MEM_BOUND or UNIT_BASE and UNIT_BOUND would be mapped into the lower 64 Gbytes of the 64-bit FB+ address region. It would also mean that all MBus memory would be accessible from FB+ (between MEM_BASE and MEM_BOUND) and no MBus memory would be private.

Page-Memory Model

A slightly more complex memory model is defined here that allows for private memory on the MBus.

When a slave to the MBus, a 4-bit FB+ page register is used to map one of 16 4-Gbyte MBus memory regions into the FB+ 32-bit address space. Within the 4-Gbyte FB+ page, the MEM_BASE and MEM_BOUND registers contained in the TI FB+ chipset point to local public memory on the MBus. Other addresses within the page but outside of MEM_BASE and MEM_BOUND are remote addresses and are transported to FB+ via the HIF.

When a slave to the host interface, the MBus-page register is used to map the incoming 32-bit FB+ address to one of 16 4-Gbyte MBus memory regions. Within the 4-Gbyte MBus page, the MEM_BASE and MEM_BOUND registers contained in the TI FB+ chipset point to memory on the MBus, which can be accessed from FB+.

If the FB+ and MBus page are kept the same, the addresses outside the page are private (i.e., not accessible by the other bus).

Figure 3 shows the memory mapping between MBus and HIF for the page-memory model.

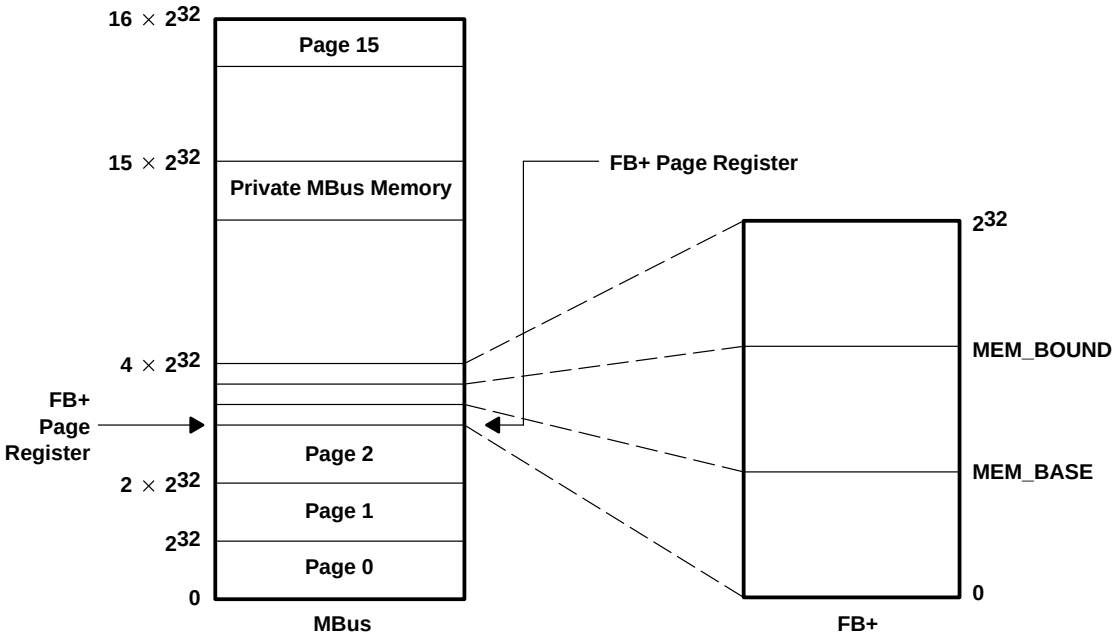


Figure 3. MBus/FB+ Memory Map

Datapath

Considerations

There are several considerations when designing the datapath interface between the MBus and the host interface.

- Since MBus runs at 40 MHz nominally and the HIF at 20–25 MHz, a FIFO is needed to synchronize the two different time domains.
- MBus has a big-endian datapath and requires that words and half words be word and byte aligned. The HIF has no endian preference with the exception of big-endian access to FB+ CSR space.
- MBus always multiplexes address and data on 64 signals; the HIF has demultiplexed address and data when the data width is 32 bits, and multiplexed address and data when the data width is 64 bits.
- MBus does 1-, 2-, 4-, and 8-byte nonburst (word) and 16-, 32-, 64-, and 128-byte burst transactions while the HIF does 1-, 2-, 3-, and 4-byte nonburst (single) and 8-, 16-, 32-, and 64-byte burst transactions. MBus bursts are always of data width 64. HIF bursts can be of data width 32 and 64.
- MBus has no parity protection on its address/data lines. The HIF address/data lines do have parity; however, the TI chipset can generate parity internally and pass it on to FB+ when sourcing data to FB+.

Taking the above considerations into account requires the use of two bidirectional FIFOs capable of being clocked at 40 MHz. These FIFOs need to be 32 bits wide and 64 words deep. They need empty/full flags, clock enables, and port-direction control. A byte-swap function within a 16-bit word also is required. The TI SN74ABT3614 is an ideal candidate.

Byte-Lane Mapping

FB+ systems require address invariant byte-lane mapping. This means that data byte 0 (the byte pointed to by byte address 0) always appears on FB+ AD (7:0). MBus systems require that data transfers of less than a double word (8 bytes) be aligned. Figure 4 shows byte-lane mapping between MBus, the FIFOs, the HIF, and FB+ for a 64-bit implementation only.

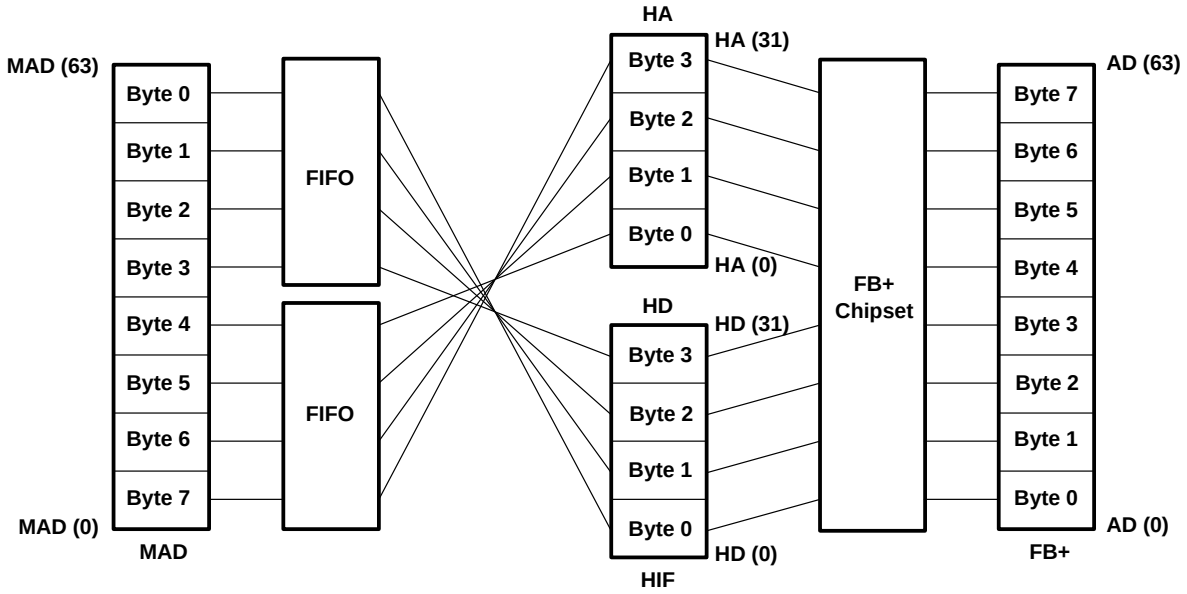


Figure 4. Byte-Lane Mapping for a 64-Bit-Only System

Since 32-bit data transfers can take place on FB+ and the HIF, quadlet (4-byte word) steering is required to correctly align 32-bit data quadlets from the HIF to those of the MBus. Quadlet steering of this type require a cross-point switch with two 32-bit ports on each end. This could be implemented with four pairs of 16-bit Widebus™ transceivers; however, this is expensive in terms of board space and datapath performance. A FIFO with the ability to swap bytes within 16-bit words performs the quadlet-steering function with the byte lanes wired as shown in Figure 5.

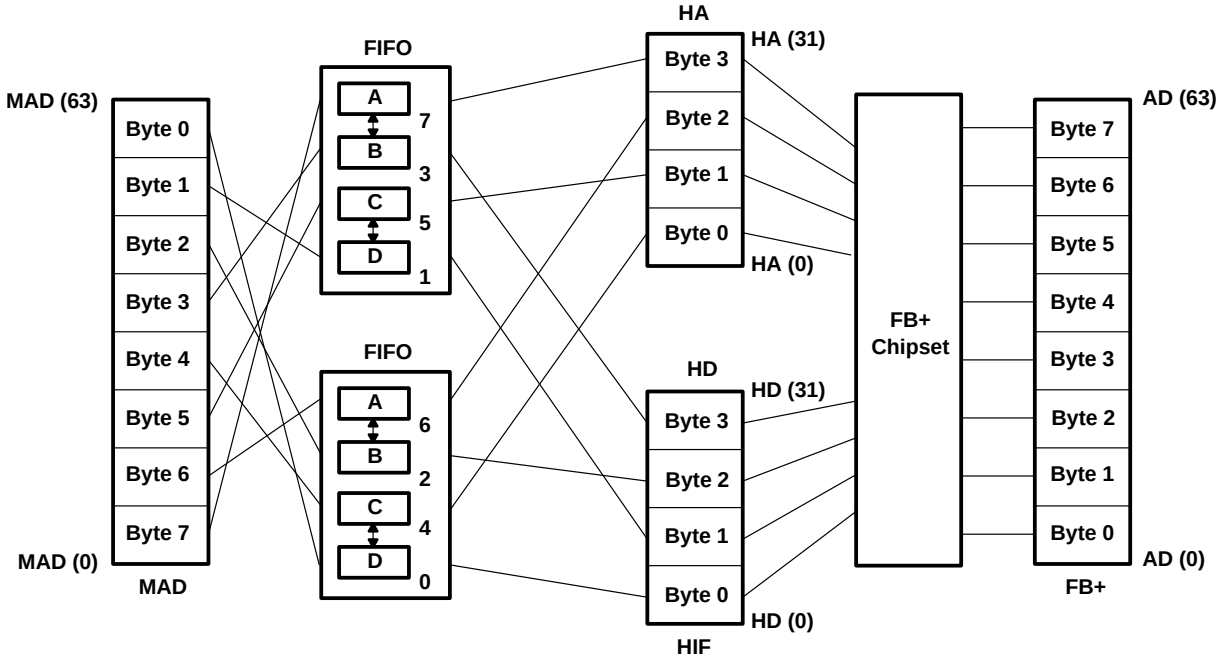


Figure 5. Byte-Lane Mapping for a 64-/32-Bit System

Widebus is a trademark of Texas Instruments Incorporated.

An example of this quadlet steering in action is the following scenario: the TI FB+ chipset is a slave to a single-beat DW64 write transaction on FB+ (8 bytes). The chipset's 64-bit HIF enable bit is not set; therefore, it performs 4-byte-wide, 8-byte burst writes onto the HIF. This HIF burst write contains two 4-byte data phases. The first data is an even-address quadlet and needs to be steered to MAD (63:32). The second data is an odd-address quadlet and needs to be steered to MAD (31:0). Figure 6 shows the datapath for the odd-address quadlet.

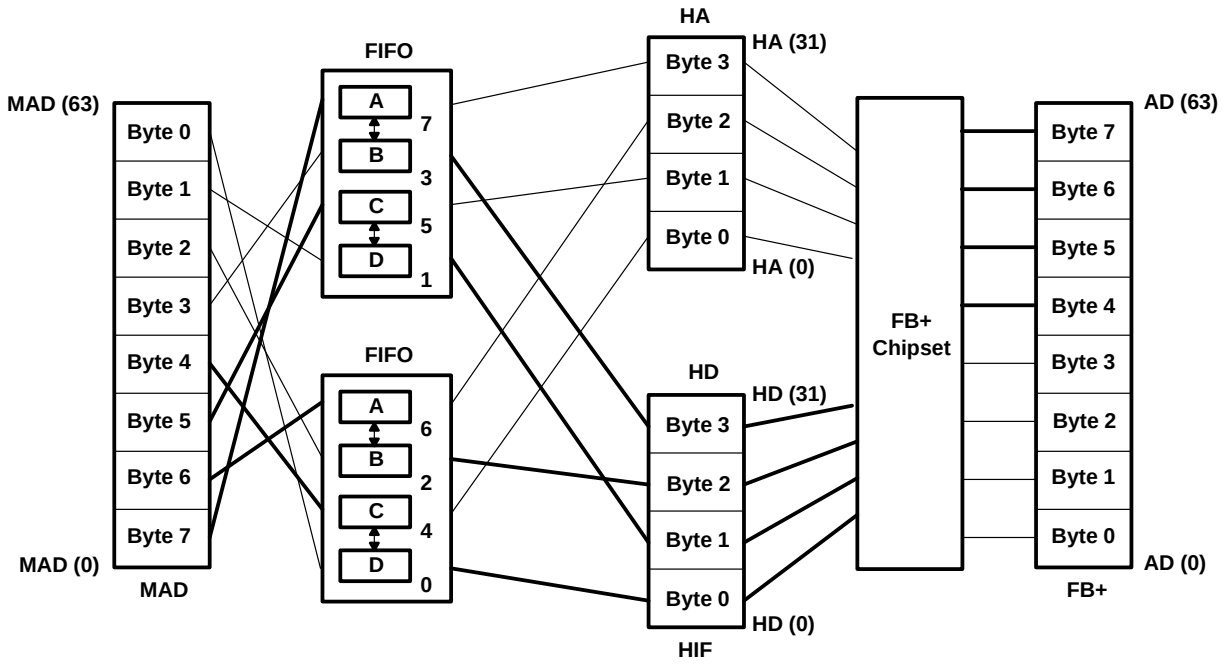


Figure 6. Byte-Lane Mapping Showing Path Taken by Address Quadlet on HD

Control Logic

The MBus controller (MBC) and host interface controller (HIFC) are synchronous state machines that operate in the two clock domains. They handle the protocols of their respective buses. They also handle arbitration protocols when bus mastership is required. These controllers drive the latch enables, 3-state controls, and FIFO control signals of the command, address, and datapath sections of the L1 bridge. They are responsible for coordinating data flow between the two buses that operate at different data rates by using the master hold-off and slave-wait capabilities of their respective buses. They also are responsible for manipulating arbitration protocols in response to locked-transaction requests.

Details of the implementation of these bus controllers is outside the scope of this application report. An example of an MBus-initiated word-read transaction resulting in an HIF single read is shown in Figure 7.

The L1 bridge is a slave to MBus and a master on the host interface. It responds to an MBus address as a selected slave and then arbitrates for HIF mastership.

Likewise , an example of an MBus-initiated word-write transaction resulting in an HIF single write is shown in Figure 8. In this case, the MBus write is acknowledged before the HIF write is complete. If another write occurs to the L1 bridge before the HIF transaction is complete, an MBus relinquish and retry operation must be performed to back off the MBus master until the HIF transaction is finished.

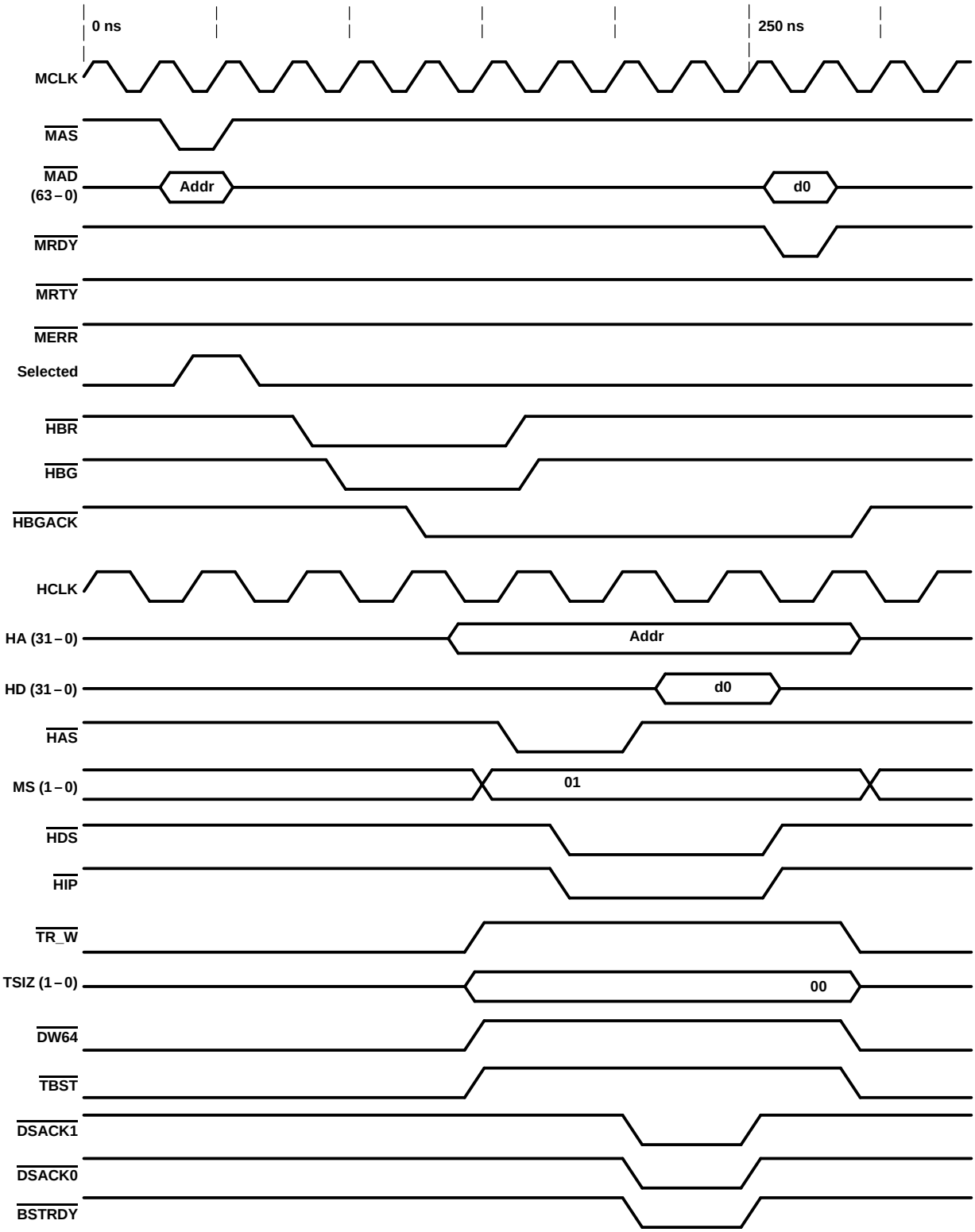


Figure 7. MBus-Word Read to HIF Single-Read Transaction

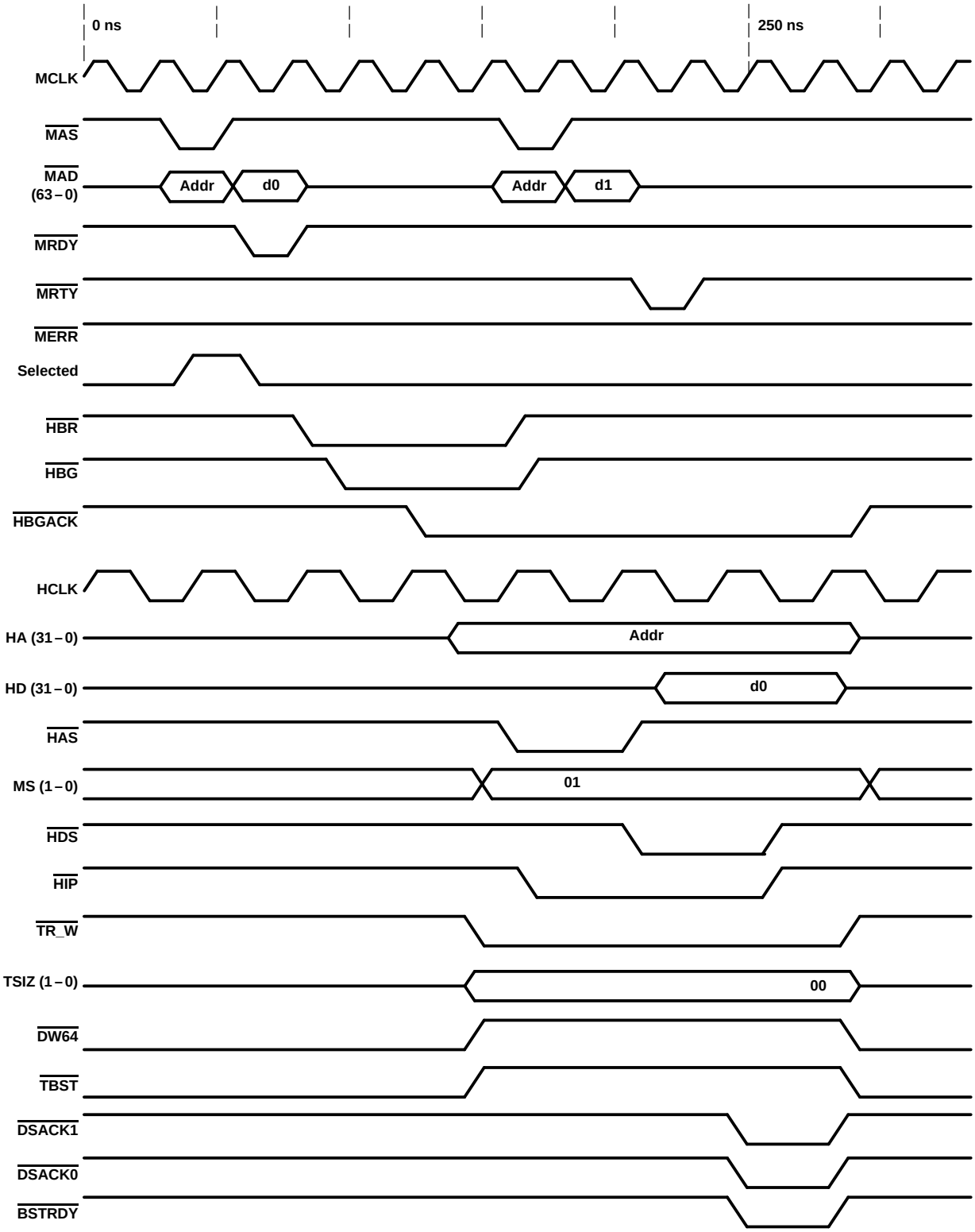


Figure 8. MBus-Word Write to HIF Single-Write Transaction

Summary

Methods of connecting the SPARC MBus to the TI FB+ chipset's host interface are explored. Level 1 MBus transactions are mapped to HIF transaction by the bridge logic. The HIF transactions are then mapped to FB+ I/O transaction by the chipset. Direct- and paged-memory mapping are described. Techniques used to implement a 64-bit-only datapath and a datapath that has a dynamically configurable 32-bit MBus/32-bit HIF or 64-bit MBus/64-bit HIF also are described. Finally, the state-machine controller's task is summarized and example transactions directed by the controllers are shown.