Implementing a Low Cost PCI Bridge and Memory Controller

Revision 1.0
December 1999

Revision History

Date	Revision	Description
12/30/99	1.0	Initial Release

Information in this document is provided in connection with Intel products. No license, express or implied, by estoppel or otherwise, to any intellectual property rights is granted by this document. Except as provided in Intel's Terms and Conditions of Sale for such products, Intel assumes no liability whatsoever, and Intel disclaims any express or implied warranty, relating to sale and/or use of Intel products including liability or warranties relating to fitness for a particular purpose, merchantability, or infringement of any patent, copyright or other intellectual property right. Intel products are not intended for use in medical, life saving, or life sustaining applications.

Intel may make changes to specifications and product descriptions at any time, without notice.

Designers must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." Intel reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them.

The 82559ER may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

Contact your local Intel sales office or your distributor to obtain the latest specifications and before placing your product order.

.

Overview

Embedded systems frequently do not require a PCI local bus. This paper addresses embedded designs where 82559ER Fast Ethernet PCI Controller is a compelling LAN interface solution, but adding a PCI bus seems too complicated or expensive.

Required PCI Command Support

Table 1 lists all the possible PCI bus command types and denotes which ones must be implemented for an 82559ER interface. The following discussion explains the commands and describes ways to reduce the complexity (and thus the cost) of the PCI bridge/memory controller. It is assumed that the designer has absolute programming control over the operating system and knows what hardware will be placed on the PCI bus (i.e., no bus connectors) so that strict PCI compliance is not necessary.

PCI bus masters, including the 82559ER Ethernet Controller, issue bus commands based on an assumption that data in memory is prefetchable. However, data prefetching is not a strictly required implementation in a PCI target.

PCI Command Type	Initiated by the Low Cost Bridge/Controller as a PCI Bus Master	Initiated by the 82559ER Fast Ethernet Controller as a PCI Bus Master
Interrupt Acknowledge		
Special Cycle		
I/O Read	Optional	
I/O Write	Optional	
Reserved		
Memory Read	X	X
Memory Write	X	X
Configuration Read	X	
Configuration Write	X	
Memory Read Multiple		X
Dual Address Cycle		
Memory Read Line		X
Memory Write and Invalidate		Optional

Table 1. PCI Bus Command Types for Systems with an 82559ER Ethernet Controller

I/O Read and *I/O Write* cycles can be used to read/write control and status registers in the 82559ER device and the EEPROM space. The use of I/O mapping is optional for the

82559ER Fast Ethernet Controller. Since many embedded systems do not use separate memory and I/O spaces, it is recommended not to use I/O mapping at all.

Memory Read and *Memory Write* commands are the most fundamental PCI memory commands and consist of burst accesses of varying lengths. The 82559ER will issue Memory Read cycles to transmit control blocks in memory and it will issue Memory Write cycles to command blocks, receive frame descriptors, receive buffer descriptors, statistics counters and receive memory data buffers. The 82559ER device is capable of long burst length transfers, but TCP/IP protocol segmentation leads to relatively small data transfers in many networks.

Memory Read Multiple cycles are like Memory Read cycles except that they are encoded to indicate that the 82559ER device expects to retrieve data beyond the end of the first cache line in a data cache. The 82559ER Memory Read Multiple is a performance enhancing command used in accesses to receive data buffers. However, caching is too complex to implement for a low cost PCI bridge and memory controller. Simply alias this command type to the Memory Read cycle type.

Memory Read Line cycles are also a performance enhancement as they are encoded to access an entire cache line. The 82559ER device will issue this cycle type for accesses to command blocks, transmit block descriptors, receive frame descriptors and receive buffer descriptors. Once again, caching should be avoided in a low cost design. The PCI specification only requires that the target accept the command and behave as if it is a Memory Read.

Memory Write and Invalidate is also a command that improves performance when a write-back caching bridge/memory controller is present. When the command is enabled, it is used to write receive packet data into buffers. Memory Write and Invalidate commands can be turned off in the 82559ER Ethernet Controller by setting the Cache Line Size bits in the Cache Line Size Register to any value other than 8 or 16. When the feature is disabled, the 82559ER device will issue Memory Writes instead.

The 82559ER Ethernet Controller has a feature which enables it to terminate read accesses on a cache line boundary whenever possible as a performance enhancement. This feature also relies on the programmed value of the Cache Line Size bits. For best performance in a non-caching system, be sure to disable this capability by writing 0 to the Read Align Enable Bit.

Considering Performance versus Simplicity

Many choices can be made regarding system performance versus complexity based on which devices will share the PCI bus with the 82559ER Ethernet Controller and how much memory bandwidth they need. The basic design consists of address translation blocks, buffer FIFOs, DMA and arbitration logic. A reasonable design goal might be a controller cost comparable to the PCI devices to be supported (i.e., the 82559ER Ethernet Controller).

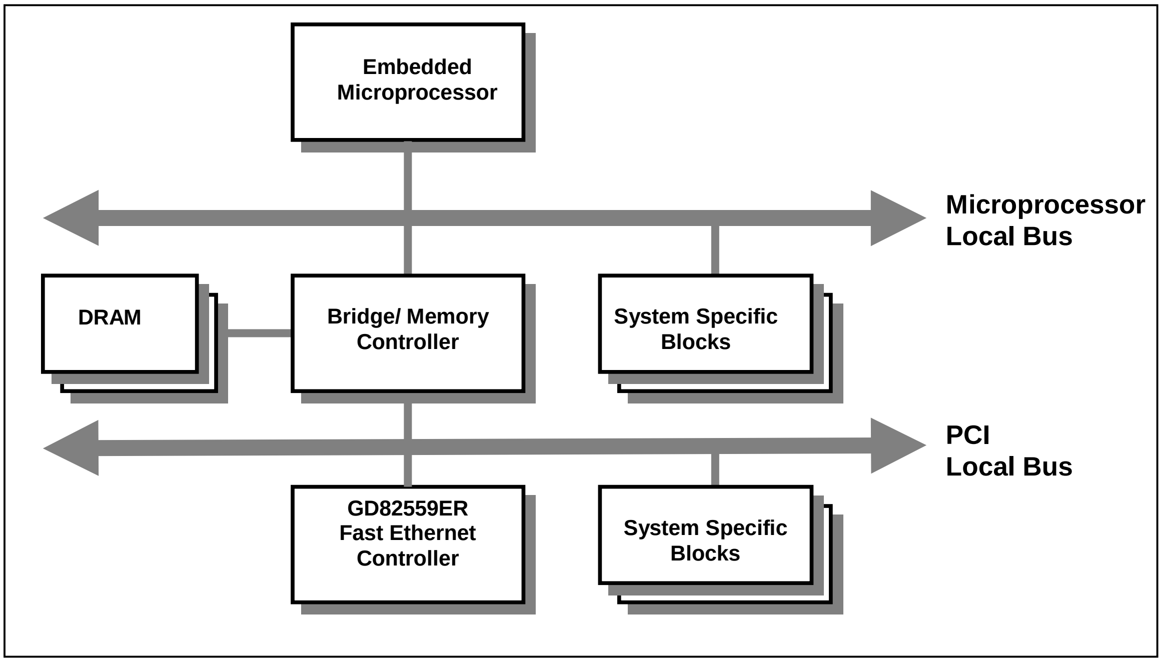

Figure 1. Generalized Embedded Design with an 82559ER Ethernet Controller

Keep in mind that recent PC chipsets do not include a cached memory bridge/controller. Unless you have several PCI masters accessing memory and need very high performance, chances are your design does not require cached memory either.

When multiple PCI bus masters exist on the PCI bus, master latency timers are used to guarantee fairness. The 82559ER device has such a latency timer. In a system where the bridge/memory controller and the 82559ER Ethernet controller are the only PCI masters, the vast majority of the data flow will be initiated by the 82559ER device and it may not be necessary to implement a master latency timer in the bridge. If needed, a leftover timer may be available elsewhere in the system.

One way to reduce cost and complexity in a bridge/memory controller is to avoid implementing parity generation and checking on the PCI bus. Without parity, the system will not comply with the PCI specification, but compliance may not be required in a “controlled” design. Reporting of parity errors on the PERR# and SERR# pins can be turned off in the 82559ER device, but if data parity is missing, the Ethernet controller will constantly report a data parity error in its PCI Configuration Status Register. However, the 82559ER Ethernet controller will not terminate any bus cycles due to parity errors and software can merely ignore the Detected Parity Error Bit.

Other specific design recommendations that can be followed to implement a low cost bridge/memory controller include:

- Rely on the large 3 KByte transmit and receive FIFOs in the 82559ER device instead of implementing large buffers in the bridge/memory controller.
- Keep the DRAM bus width to 32 bits.
- Minimize the number of DMA channels.
- Don’t provide for concurrent main processor and PCI device access to memory.
- Minimize the number of devices on the PCI bus.
- Avoid the use of PCI expansion card slots.
- Minimize powerdown logic in the bridge/memory controller.

Resorting to Off-the-Shelf Solutions

If your PCI bus control requirements exceed what you can implement in a low cost design, you can consider an off-the shelf, standard PCI controller. PCI bridge/memory controller devices are available to support members of the following microprocessor families: Intel Architecture, Intel i960 ®, QED RM52xx (MIPS*), IDT 4xxx/5xxx (MIPS), NEC VR4300 (MIPS), PowerPC* (both IBM and Motorola), Motorola 68K* and Hitachi SuperH*.

Another alternative would be to use a microprocessor with a built-in PCI bridge/memory controller, such as Intel’s i960VH Embedded-PCI Processor. The 80960VH processor is ideal for many embedded computing applications such as VPN appliances and Ethernet expansion boards.

Conclusion

Adding a full-fledged PCI bridge and memory controller to an embedded processor system may seem like a daunting task. However, designers will find it possible to considerably minimize the PCI local bus functionality and still adequately support the 82559ER Ethernet Controller.

References

1. Altera Corporation, “PCI Bus Applications in Altera Devices,” Application Note 41, Altera Corp., 1995, Version 1.

2. Jim McManus, “Using FPGAs as a Flexible PCI Interface Solution,” Xilinx, Inc., 1998, Version 2.0.

3. Jato Technologies, “Optimizing Network Interface Cards for Operation in a Standard High Volume Server,” Jato Technologies, 1998.

4. PCI Special Interest Group, “PCI Local Bus Specification,” 1999, Rev. 2.2.