

Implementation of 802.11a/b/g in Microprocessor Systems Using a QuickPCI Bridging Solution



••••• QuickLogic® White Paper

Abstract

This white paper describes how to add 802.11a/b/g wireless connectivity to a microprocessor that does not have an integrated PCI interface. Wireless connectivity can easily be added to your system using a bridging solution based on QuickLogic's new PCI 2.3-compliant devices. The following sections discuss the common interface available on wireless chips, the method of bridging wireless chips to processors, and system design considerations.

Introduction

802.11a/b/g has been established as the most common standard for wireless data transfer. Originally designed for portable personal computers, 802.11a/b/g is moving into other application areas such as video surveillance, IP phones, smart phones, point-of-sales terminals, portable media players, and many more.

The predominant interface on the 802.11a/b/g chipsets currently on the market is the PCI bus, the standard bus in PC systems. In portable or embedded applications PCI is usually used with the same signalling but a different form factor. Many 802.11a/b/g modules are available in the miniPCI or Carbus format. Newer embedded microprocessors frequently have an integrated PCI interface that allows easy connection to standard 802.11a/b/g chipsets. However, many legacy and low cost processors do not have an integrated PCI interface. Instead, they support a processor specific local bus or a simple memory interface (e.g., SDRAM). Currently there are no 802.11a/b/g chips available that can directly interface to these processor-specific local busses. To add wireless connectivity to a system with a legacy processor, a bridge device is required. This bridge device allows leveraging the substantial investment companies have made in their original hardware, software, and in intellectual property. QuickLogic bridging solutions are ideally suited for portable wireless applications. The new FPGA architecture of the Eclipse II and QuickPCI devices allows designers to meet the power requirements of these portable applications that usually have a very tight power budget.

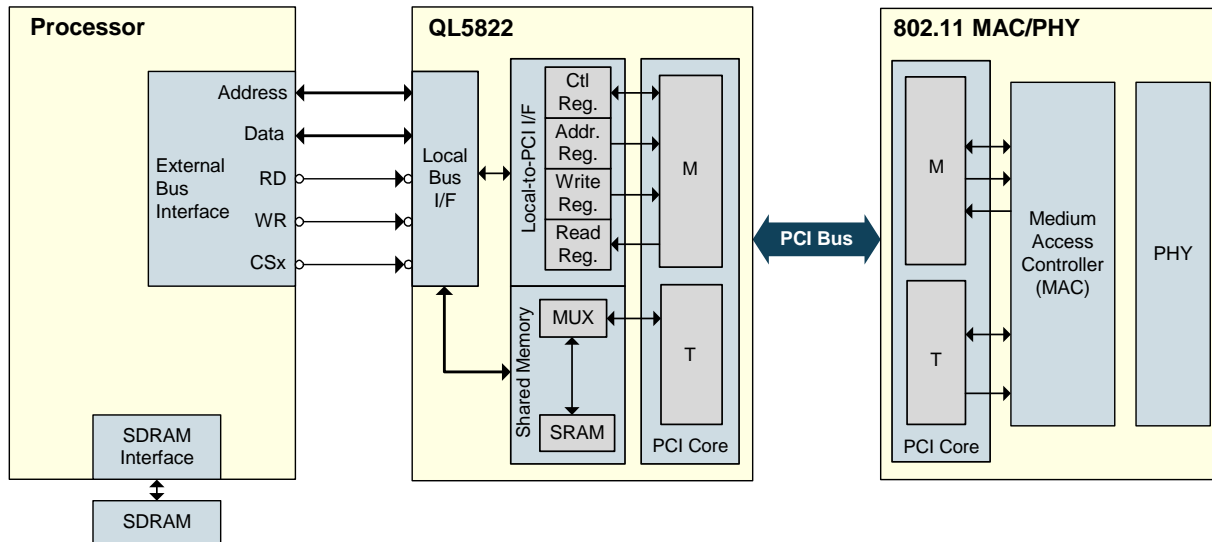
The following sections describe two ways to implement a bridge between an embedded processor and an 802.11a/b/g chipset. Based on the interfaces available on the processor and bandwidth consideration the designer can choose the architecture that best fits into their system.

For both of these implementation options, the wireless chipset is connected via PCI to the bridge device. QuickLogic QuickPCI devices contain a fixed-function PCI core embedded in a programmable fabric. The programmable fabric provides the flexibility necessary to bridge between different local bus architectures and the embedded PCI core. For details about the architecture of the newest QuickLogic PCI devices, see http://www.quicklogic.com/images/QL58x2_Family_DS.pdf.

Option I – Direct Bridge Implementation with QuickPCI

Figure 1 illustrates Option I: Direct bridge implementation with QuickPCI.

Figure 1: Processor to 802.11a/b/g Bridge (Option I)



In this implementation only the local bus of the processor is connected to the bridge device. The implementation in hardware is simple and straightforward, however the designer needs to carefully consider potential bottlenecks in the data throughput. If the local processor bus supports an external device to master the bus and write data directly into the processors memory, the bridge can forward any DMA transfer executed by the wireless chipset to the system memory.

However, many embedded processors do not support this type of direct access to the main memory. To get the data in and out of these devices the processor needs to actively read or write on the local bus. Since the 802.11 a/b/g chipsets are built to support DMA on the PCI bus, the bridge would need to be configured as a bi-directional buffer device allowing the wireless chipset to access the buffer via PCI DMA while the processor accesses the other side via its local bus protocol.

To synchronize the local bus and PCI bus data streams, data must be buffered within the bridge device. To achieve the desired data throughput, two closely related design parameters must be considered:

- The size of the buffer RAM within the PCI bridge device.
- The latency time on the local bus between a request for a transfer on the local bus and the execution by the microprocessor.

Based on the throughput requirement defined by the application and a latency time given by the software architecture (e.g., interrupt response time), the required RAM size can be calculated and the appropriate QuickLogic solution can be selected accordingly.

Table 1 lists typical wireless applications and their bandwidth requirements.

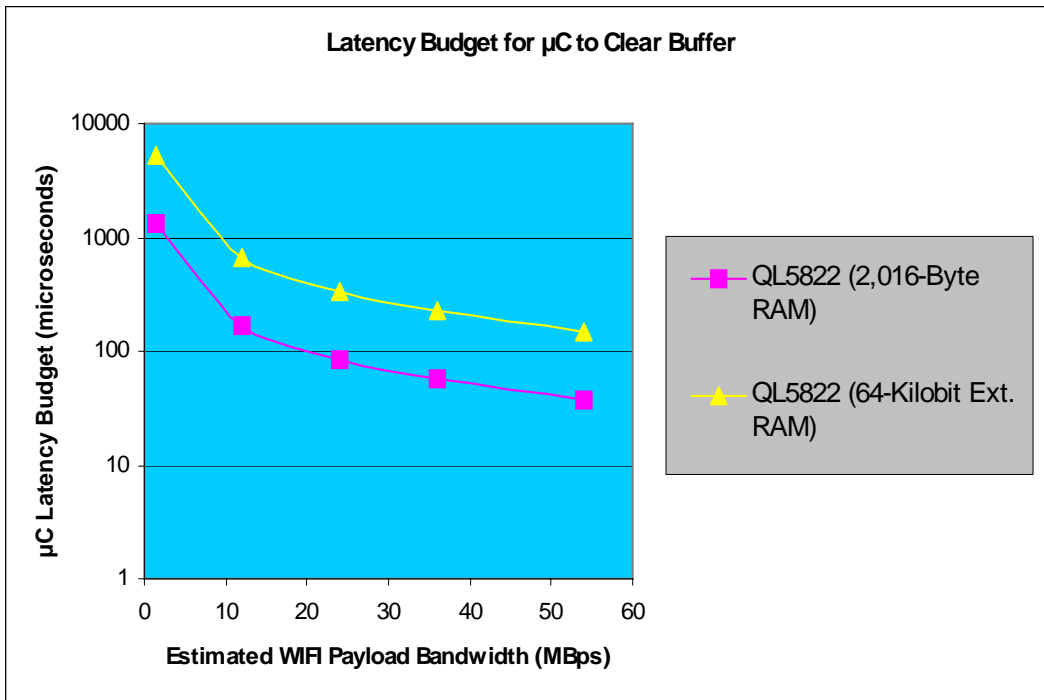
Table 1: Wireless Applications and Bandwidth Requirements

Applications	Payload Bandwidth (bps)	Payload Bandwidth (byte/sec.)
MPEG2 video	9.8 M	1.225 M
MPEG4	2.254 M	281.8 K
MPEG4-AVC/H.264 video	1 M	128 K
5-channel audio	320 K	40 K
MP3 (stereo and CD quality)	128 K	16 K
ISDN (per B channel)	64 K	8 K
AVI video streaming	64 K	8 K

For Option I, QuickLogic offers different bridging alternatives that can be tailored to application bandwidth requirements and processor latency time. The first and most straightforward option is a design based on a QL5822 QuickPCI device programmed with a local interface. An internal RAM buffer is used to buffer the data between the wireless module and the processor. If the latency/throughput requires more memory in the bridging device, external SRAM can be attached to the QL5822 device to achieve the necessary throughput under any conditions.

Figure 2 shows the maximum allowed processor latency time for different bridging solutions and throughput requirements.

Figure 2: Latency Budget for Different QuickLogic Solutions and Throughput

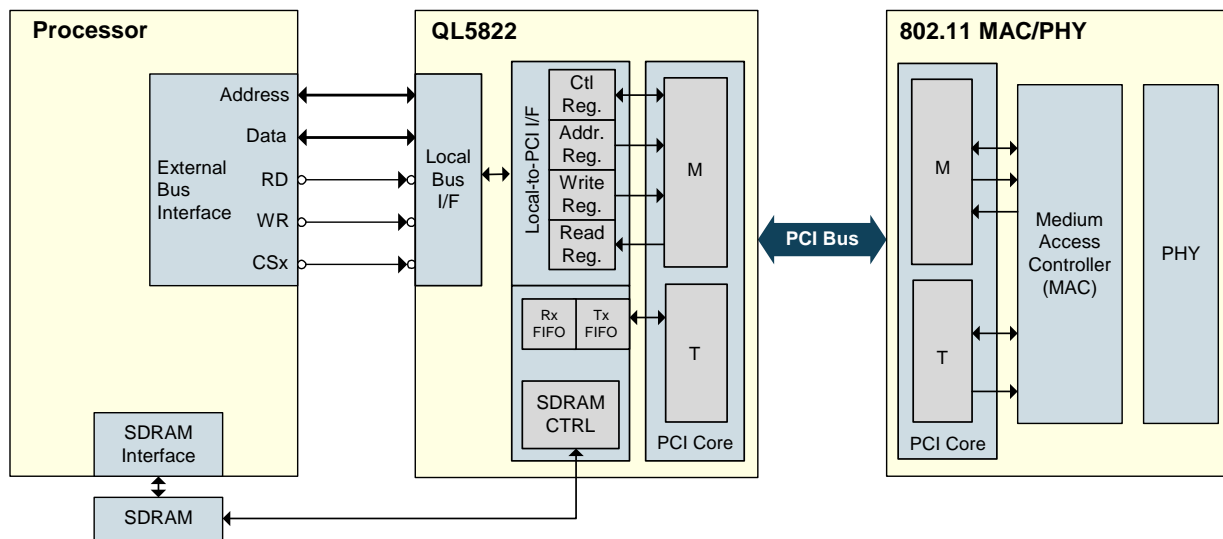


This first bridging implementation option requires a careful analysis of the bandwidth requirements and its maximum throughput depends on the software response time for each data packet transfer. Since 802.11a/b/g chipset have originally been built for a PC environment, drivers for a different architecture are usually not available from the vendors of the wireless devices requiring the design team to create custom drivers and carefully optimize the software architecture.

Another important aspect to consider is the DMA scheme usually implemented by the 802.11a/b/g devices. The wireless device contains a DMA engine which is programmed by pointing to a list of descriptors. Each descriptor contains the start address and length of a data block to be transferred by the DMA engine. These descriptors are usually stored somewhere in the main memory of the host which is directly accessible to the DMA engine via PCI. In the first bridging implementation the descriptors need to be stored in the same buffer that is already used for storing the data packets. For straight forward software implementations it is necessary to store at least one or two data packets for each direction along with the descriptors in the buffer memory. This basic buffering can be achieved with the internal memory of QL5822. To reduce the interrupt load and to further simplify the software effort a bigger external SRAM buffer can be used.

Option II – SDRAM Bridge Implementation with QuickPCI

Figure 3: SDRAM Bridge Implementation with Quick PCI (Option II)



Some local processors do not allow an external device to master the bus and write data directly into the processor memory, however, they do support external mastering of the SDRAM. For these processors the bridge can be designed in a different way. The local bus of the processor and the SDRAM bus are connected to the bridge device. The hardware implementation is a little more complicated since the SDRAM bus must now be connected to an additional device and the bridge must support the SDRAM access protocol. However, this architecture has several advantages over Option I.

First and foremost, the wireless driver implementation is much simpler. The architecture is basically the same as the common PC architecture. The 802.11a/b/g chipset can transfer data using DMA via the transparent bridge directly into the main memory of the processor. The descriptors for the DMA transfer and a large number of data packets can be stored within the main processor memory without being limited by the size of

the buffer memory available to the bridge. Therefore, standard PC software drivers can be reused without major modifications. The second advantage is that the processor is not burdened with the handling of data packets transferred in and out of the buffer in the bridge. Since all basic data transfers are handled in hardware, interrupt response time and stacking of interrupts is not a concern. Finally, the buffer size is also not a concern, since the bridge will just simply delay any PCI transaction until the SDRAM bus is available. Delays on the PCI bus have no impact on the system throughput considerations since the maximum 802.11a/b/g bandwidth of 54 megabits per second is much smaller than the available PCI bandwidth of 132 megabytes per second.

QuickLogic recommends Option II for all processors that support external mastering of the SDRAM bus.

Conclusion

QuickLogic offers several custom solutions that can bridge an 802.11a/b/g PCI device to an existing microprocessor device. Depending on the data throughput requirements and the latency time of the microprocessor, the optimal solution can be selected. The programmable fabric of QuickLogic devices provides bridging between any microprocessor device with a generic local bus and a PCI 802.11a/b/g device. Additionally, QuickLogic low power PCI devices enable wireless bridging solutions for many types of portable power sensitive applications.

Contact the QuickLogic support team at <http://www.quicklogic.com/support> or your local sales representative to see how quickly a solution can be tailored to your specific processor bus and application needs.

Contact Information

Phone: (408) 990-4000 (US)
(905) 940-4149 (Canada)
+(44) 1932-57-9011 (Europe)
+(852) 2567-5441 (Asia)

E-mail: info@quicklogic.com

Sales: America-sales@quicklogic.com
Europe-sales@quicklogic.com
Asia-sales@quicklogic.com
Japan-sales@quicklogic.com

Support: www.quicklogic.com/support

Internet: www.quicklogic.com

Revision History

Revision	Date	Originator and Comments
Rev. A	March 2004	Bernhard Andretzky and Kathleen Murchek
Rev. B	October 2004	Bernhard Andretzky and Kathleen Murchek
Rev. C	August 2005	Bernhard Andretzky and Kathleen Murchek
Rev. D	January 2009	Kathleen Murchek

Notice of Disclaimer

QuickLogic is providing this design, product or intellectual property "as is." By providing the design, product or intellectual property as one possible implementation of your desired system-level feature, application, or standard, QuickLogic makes no representation that this implementation is free from any claims of infringement and any implied warranties of merchantability or fitness for a particular purpose. You are responsible for obtaining any rights you may require for your system implementation. QuickLogic shall not be liable for any damages arising out of or in connection with the use of the design, product or intellectual property including liability for lost profit, business interruption, or any other damages whatsoever. QuickLogic products are not designed for use in life-support equipment or applications that would cause a life-threatening situation if any such products failed. Do not use QuickLogic products in these types of equipment or applications.

QuickLogic does not assume any liability for errors which may appear in this document. However, QuickLogic attempts to notify customers of such errors. QuickLogic retains the right to make changes to either the documentation, specification, or product without notice. Verify with QuickLogic that you have the latest specifications before finalizing a product design.

Copyright and Trademark Information

Copyright © 2009 QuickLogic Corporation. All Rights Reserved.

The information contained in this document is protected by copyright. All rights are reserved by QuickLogic Corporation. QuickLogic Corporation reserves the right to modify this document without any obligation to notify any person or entity of such revision. Copying, duplicating, selling, or otherwise distributing any part of this product without the prior written consent of an authorized representative of QuickLogic is prohibited.

QuickLogic and QuickPCI are registered trademarks, and Eclipse II and the QuickLogic logo are trademarks of QuickLogic. Other trademarks are the property of their respective companies.