Freescale Semiconductor
Application Note

Document Number: AN3373
Rev. 0, 04/2007

PowerQUICC™ II Pro (MPC83xx) PCI Agent Initialization

by: David Smith
Field Application Engineering
Raleigh, NC

In many designs, the PowerQUICC™ II Pro (MPC83xx) operates as a peripheral component interconnect (PCI) agent device that enables system add-in functionality as a PCI card. The operating software for these processors can reside within local memory or in PCI memory space, though local memory is preferred from a system performance standpoint. The initialization code can reside in a locally-attached flash memory device that performs enough system initialization to load the software and then jump to the operational software. However, to simplify system management, it is preferable to initialize and load the PowerQUICC II pro device from the PCI host, eliminating the need for the flash memory device and providing a centralized means to manage the initialization code.

This application note describes the steps required to initialize an MPC83xx device operating as a PCI agent. It compares initialization from the PCI host with initialization from the on-board e300 processor. After an overview of the initialization process, it discusses key points of the architecture, hardware reset, minimum initialization required to run software, and system implications. These points apply not only to the MPC834x family but also to any PCI agent-attached MPC83xx product, with perhaps minor adjustments.

Contents

1 PowerQUICC II Pro PCI Agent Mode

Figure 1 shows a block diagram of a typical system with the MPC83xx device connected as a PCI agent feature card. The run-time software for the agent card initially resides in the PCI host system complex, perhaps physically in flash memory (either on card or as a pluggable flash module), in a hard drive connected to the host complex, or in a central server directly accessible by the host via a network connection. With this approach, the user has the advantage of managing the PCI agent software from a central point and the amount of on-card non-volatile storage for each adapter card is reduced. Either the on-board PCI agent processor or the PCI host can initialize the PCI agent card. Initializing with the PCI host allows the flash memory device to be removed, which can reduce per card cost, simplify the hardware design, and simplify management of the initialization code.

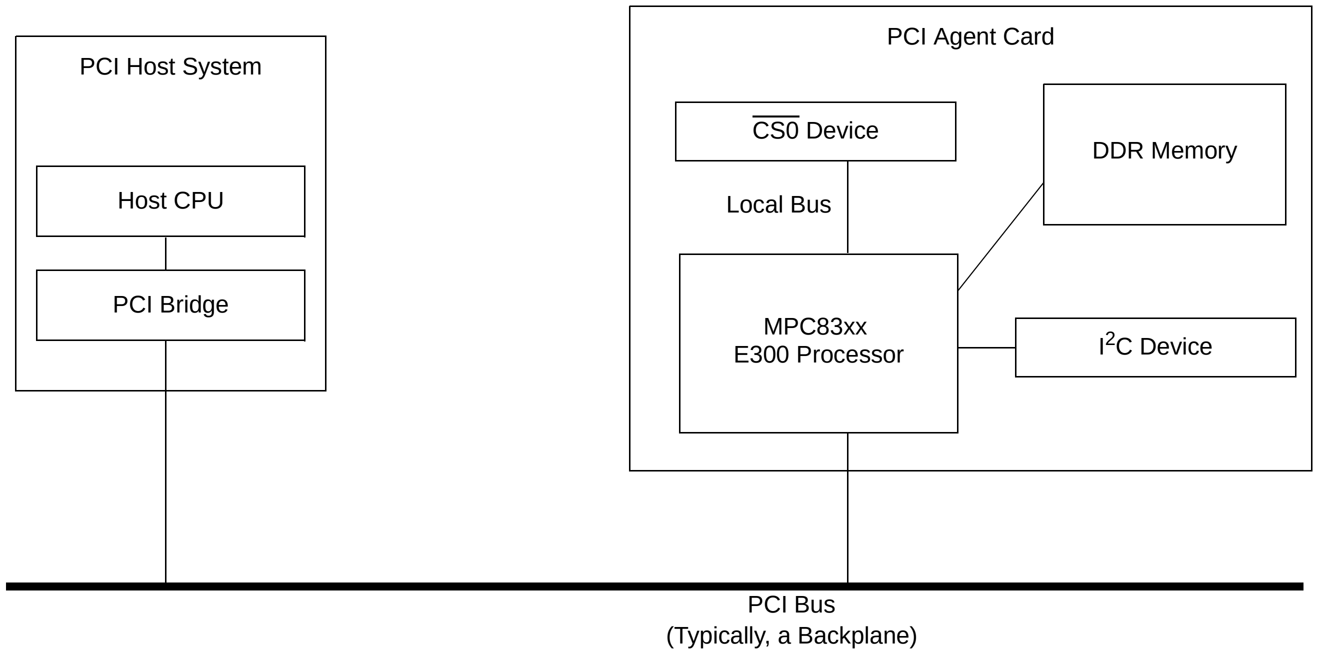

Figure 1. Typical PCI Host/Agent System

1.1 Initialization Flow

The general initilization flow of a PCI agent card is similar regardless of which processor performs the tasks (PCI host or on-board e300 core). Table 1 shows the similarities and differences in the initialization flow for the two initialization sources.

Table 1. Comparison of Initialization Steps

On-Board Initialization, PCI Host Software Load	PCI Host Initialization, PCI Host Software Load
1. The hardware reset configuration word is read; it must be configured to enable the e300 core.	1. The hardware reset configuration word is read; it must be configured to disable the e300 core.
2. PCI configuration cycles are locked (default).	2. PCI configuration cycles are enabled (either by a reset sequence or the boot loader).
3. The IMMR address is set to default in local memory space, relocate as necessary.	3. The PCI host initializes the adapter PCI configuration space. At a minimum, the PIMMR must be initialized.
4. The internal e300 processor initializes the memory controller, PCI inbound/outbound windows, and (optionally) PCI configuration space.	4. The PCI host initializes the memory controller and PCI inbound/outbound windows.
5. The internal e300 processor enables PCI configuration cycles and the PCI host initializes PCI configuration space (if not done previously).	5. Not applicable.
6. The PCI host writes run-time software into agent card memory.	6. The PCI host writes run-time software into agent card memory.
7. The PCI host notifies the on-board e300 core to run the software.	7. The PCI host enables the core, which runs the software.

1.1.1 Reset Considerations

The reset configuration word defines operational parameters critical to initialization. There are three different methods to define the reset configuration word:

- From the device on $\overline{\text{CS0}}$.
- From a serial EEPROM connected to the I^2C.
- Select one of the hard-coded default options.

For on-board initialization, the core must be enabled as the internal e300 processor performs the device initialization. However, if the PCI host is to initialize the board, the core must be disabled. All the hardcoded default choices disable the core so that this option can be used only for PCI host initialization. Table 2 compares the reset options and restrictions:

Table 2. Specifics of Reset Configuration Source

Reset Configuration Source	On-Board Initialization	PCI Host Initialization

Chip Select 0 Source	COREDIS = 0	COREDIS = 1
		BOOTSEQ = 01 or 10

I^2C Source	COREDIS = 0	COREDIS = 1
		BOOTSEQ = 01 or 10

Default hardcoded	Not applicable	Core is disabled, and CFG_LOCK in PCI FCR is set

The reset modes also affect the minimum hardware required. For PCI host initialization, the CFG_LOCK in the PCI FCR must be set so the host can access the card. If the default hardcode method is used, this is done at reset,. However, if the other methods are used, the boot sequencer must be used to set the CFG_LOCK in the PCI FCR within the MPC83xx. Using the boot sequencer requires a serial EEPROM to be connected to the I^{2}C. If the I^{2}C module is the source of the reset configuration word, the same serial EEPROM can be used for the reset configuration word and the boot sequencer information. In either case, if the chip-select 0 method is selected, either a flash/ROM or field-programmable gate array/complex programmable logic device (FPGA/CPLD) must be attached to the local bus to provide the reset configuration word. Because of these dependencies, the case requiring the minimum amount of reset/initialization support hardware is the PCI host initialization scheme using one of the default hardcoded reset configurations (no I^{2}C or $\overline{\text{CS0}}$ device required). However, the reset configuration is limited to the options defined in the reference manual for the particular MPC83xx device (typically just different clock ratios).

1.1.2 IMMR Initialization

Whether it be a separate PCI-attached host processor or the internal e300 core, the initializing processor must have the IMMR base address so that the proper MPC83xx registers are configured. In the on-board case, the only addressing in effect is the local space, so this IMMR base is the default of 0xFF400000. If this base needs to be moved to make space for another memory space (such as the boot memory space), it should be moved early on.

When the PCI-attached host performs the initialization, the IMMR base is the PCI address that maps to the internal IMMR space. The host sets this base using PCI configuration cycles by initializing the PIMMR (offset 0x10). The PCI host then initializes the agent card by writing to the area in PCI space with the base it set in the PIMMR. When the host writes the PIMMR, a 1 Mbyte inbound window is automatically defined, so the user does not need to configure, and thus consume, a PCI inbound window for this task.

1.1.3 Agent Card Hardware Initialization

Agent card hardware initialization is basically the same, regardless of whether the on-board e300 core or the external PCI host processor performs the initialization. Typically the following hardware initialization steps are performed:

1. Configure flash memory, DRAM, and local bus device chips-selects in the memory controller.
2. Configure system exceptions and the MMU/caches.
3. Configure the required system resources (timer, clocking, interrupts).
4. Configure the PCI controller, including inbound and outbound (to the MPC83xx) window definitions.

   In the on-board initialization case, the e300 core must initialize the PIMMR.
5. Configure CSB operation.
6. Configure I/O (UART, I^{2}C, TSEC, and so on).
7. An optional step specific to the agent e300 initialization case is to enable the external PCI host to perform configuration cycles. This step is necessary if the PCI host is to initialize some of the PCI configuration space (to set per adapter memory spaces, for example) and is accomplished by setting the CFG_LOCK bit in the PCI FCR.

The last step specific to the on-board e300 core initialization is to inform the PCI host that the core is ready for a software transfer. This step is accomplished in one of the following ways:

- By a defined semaphore
- Enabling PCI configuration cycles as the “last step” (that would be retried by the PCI host).
- Generating a PCI interrupt to the host. The e300 core must then go into a wait loop.

1.1.4 Starting Run-Time Software

Typically, with a PCI agent adapter the run-time software initially resides in the PCI host system. This simplifies the management of the software load(s) by placing them in a central location and allowing the main system processor to determine the appropriate software to load and run in the agent adapter.

In either scenario the PCI host can now transfer the run-time software into the memory space of the PCI adapter card (usually SDRAM). In the PCI host initialization case, the first instruction of the software must reside at the location and device specified in the BMS and ROMLOC bits in the reset configuration word (RCW). If the RCW value is one of the default hardcoded options, the boot location is 0xFF80_0000 and the device is the DDR SDRAM; the previous initialization must match this if this method is used for the RCW.

After the software is transferred, the PCI host informs the on-board e300 core to start running the software. If the PCI host performs the initialization, this is done by clearing the COREDIS bit in the Arbiter Configuration Register (ACR). If the on-board e300 performs the initialization, the PCI host can now inform the agent processor to leave its wait loop (either by semaphore or interrupt).

1.2 Recommendations

When the PCI host performs the initialization, it appears to the MPC83xx processor as a long series of PCI writes following the PCI configuration write that sets the PIMMR. The PIMMR initialization requires about 100 milliseconds in order to complete internally which holds off any update to the internal memory space. From the system perspective this appears as though the first few PCI writes are accepted but then one of the writes stalls for an excessive period. This is because the posted write buffer will fill as the PIMMR update is processed internally. To avoid this condition the PCI host can either read back the PIMMR after setting it or add a delay (~120 mS depending on system clocking) after setting the PIMMR before proceeding with the rest of the initialization of the MPC83xx.

2 Conclusion

The details of operation have been successfully implemented in a “live” system environment, and the specifics discussed here were found to be critical for proper operation. These results indicate that the MPC83xx is well suited for use as a PCI agent adapter card. The designer must determine whether to initialize the adapter card by using the on-board e300 core or the PCI host processor.   The basic steps in both cases are similar, but there are several differences that must be understood to perform the initialization successfully. The differences affect the hardware requirements of the agent adapter board (flash memory, I^{2}C) as well as the steps the initilization code must perform, wherever it is run.

3 References

MPC8349 PowerQUICC™ II Pro Family Reference Manual

THIS PAGE INTENTIONALLY LEFT BLANK

How to Reach Us:

Home Page:
www.freescale.com

Web Support:
http://www.freescale.com/support

USA/Europe or Locations Not Listed:
Freescale Semiconductor, Inc.
Technical Information Center, EL516
2100 East Elliot Road
Tempe, Arizona 85284
+1-800-521-6274 or
+1-480-768-2130
www.freescale.com/support

Europe, Middle East, and Africa:
Freescale Halbleiter Deutschland GmbH
Technical Information Center
Schatzbogen 7
81829 Muenchen, Germany
+44 1296 380 456 (English)
+46 8 52200080 (English)
+49 89 92103 559 (German)
+33 1 69 35 48 48 (French)
www.freescale.com/support

Japan:
Freescale Semiconductor Japan Ltd.
Headquarters
ARCO Tower 15F
1-8-1, Shimo-Meguro, Meguro-ku
Tokyo 153-0064
Japan
0120 191014 or
+81 3 5437 9125
support.japan@freescale.com

Asia/Pacific:
Freescale Semiconductor Hong Kong Ltd.
Technical Information Center
2 Dai King Street
Tai Po Industrial Estate
Tai Po, N.T., Hong Kong
+800 2666 8080
support.asia@freescale.com

For Literature Requests Only:
Freescale Semiconductor
Literature Distribution Center
P.O. Box 5405
Denver, Colorado 80217
+1-800 441-2447 or
+1-303-675-2140
Fax: +1-303-675-2150
LDCForFreescaleSemiconductor
@hibbertgroup.com

Document Number: AN3373
Rev. 0
04/2007