

Design Migration from the PowerQUICC™ II Family to the PowerQUICC II Pro Family

*By Christian Hammond and Jean-Philippe Giraud
Digital Systems Division
Freescale Semiconductor, Inc.*

The PowerQUICC™ II Pro family, like the PowerQUICC II family, is designed using the Freescale System on Chip (SoC) architecture, which is a modular approach to product family design. However, the PowerQUICC II Pro family—the MPC8349E, MPC8347E, MPC8343E, MPC8358E, and MPC8360E—has several enhancements over the PowerQUICC II family. As [Table 1](#) shows, these include:

- Enhanced e300 PowerPC™ core
- Dual data rate (DDR) SDRAM controller
- 10/100/1000 Ethernet MAC (TSEC—triple speed Ethernet controller).
- QUICC Engine—twin dedicated communications RISC engines (MPC8360 and MPC8358 only)
- High Speed USB 2.0
- Dual PCI (MPC8349E only)
- Integrated security engine (E versions only)

This document discusses the differences in architectures to consider when migrating from a PowerQUICC II to a PowerQUICC II Pro design.

Contents

1	Overview of Internal Architecture	3
2	PORESET	12
3	DMA Controller	15
4	e300 Core	15
5	Power Enhancements	18
6	Debug	18
7	Local Bus Controller	20
8	Double Data Rate SDRAM Memory Controller	20
9	PCI	22
10	Integrated Programmable Interrupt Controller (IPIC)	23
11	Clocks and Timers	24

Table 1. Feature Set Differences across the PowerQUICC II and PowerQUICC II Pro Families

Family	PowerQUICC II	PowerQUICC II Pro Family				
Device	MPC8280	MPC8349/E	MPC8347/E	MCP8343/E	MCP8358/E	MCP8360/E
Core	G2LE	e300	e300	e300	e300	e300
CPU Speed	Up to 450MHz	Up to 667MHz*	Up to 667MHz ¹ (TBGA) Up to 400MHz (PBGA)	Up to 400MHz	Up to 400MHz	Up to 667MHz*
Communication Processor Speed	333MHz	N/A	N/A	N/A	400MHz	400MHz
L1 I/D Cache	16KI/16KD	32KI/32KD	32KI/32KD	32KI/32KD	32KI/32KD	32KI/32KD
Memory Controller	64/32 bit SDRAM	64/32 bit DDR	64/32 bit DDR	32 bit DDR	32 bit DDR	1x64 -2x32 bit DDR
Local Bus	Yes	Yes	Yes	Yes	Yes	Yes
PCI	1 32-bit up to 66 MHz	2 32-bit up to 66 MHz or 1 64-bit up to 66MHz	1 32-bit up to 66 MHz	1 32-bit up to 66 MHz	1 32-bit up to 66 MHz	1 32-bit up to 66 MHz
Ethernet	3-10/100	2-10/100/1000	2-10/100/1000	2-10/100/1000 (MII, RGMII & RTBI only)	2-10/100/1000 (MII, RGMII & RTBI only)	2-10/100/1000 (MII, RGMII & RTBI only)
USB	Hi-speed host, device	Hi-speed host, device, and OTG	Hi-speed host, device, and OTG	Hi-speed host, device, and OTG	Full-speed host, device,	Full-speed host, device
Security	No	SEC 2.0 (E version only)	SEC 2.0 (E version only)	SEC 2.0 (E version only)	SEC 2.0 (E version only)	SEC 2.0 (E version only)
UART	Yes	Dual	Dual	Dual	Dual	Dual
I ² C	Single	Dual	Dual	Dual	Dual	Dual
SPI	Yes	Yes	Yes	Yes	Yes	Yes
Interrupt Controller	Yes	PIC-compliant	PIC-compliant	PIC-compliant	PIC-compliant	PIC-compliant

¹ e300 667 MHz core on Revision 1.1 onward.

The PowerQUICC II Pro family comprises several devices to benefit the general-purpose, industrial, and network infrastructure markets. As [Figure 1](#) and [Figure 2](#) show, there are two main PowerQUICC II Pro family architectures—the MPC834x and the MPC835x/6x.

NOTE

Not all the blocks are supported on each PowerQUICC II Pro device. Refer to the relevant device reference manual for details.

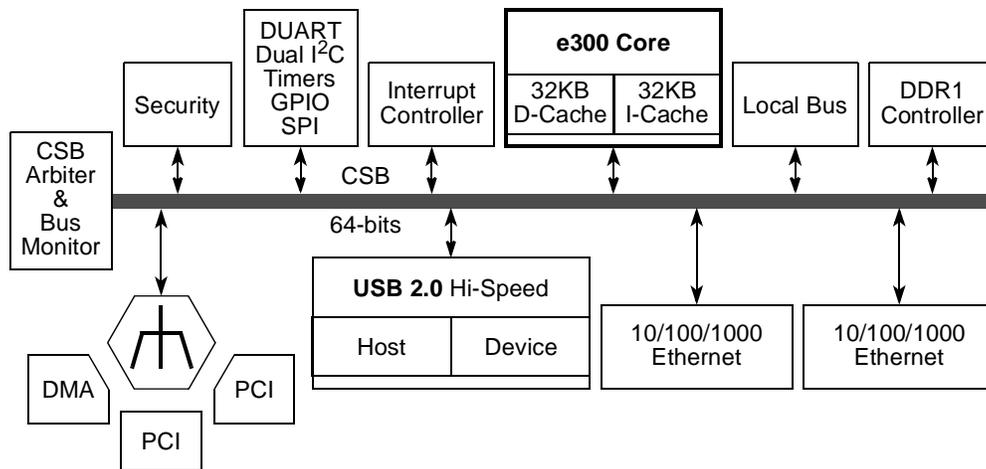


Figure 1. MPC834xE

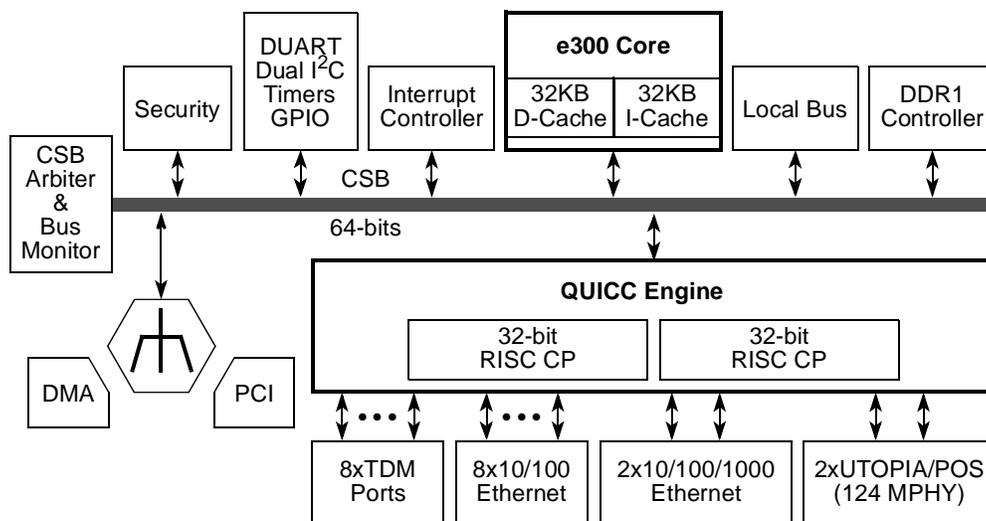


Figure 2. MPC8358E/MPC8360E

1 Overview of Internal Architecture

Figure 3 and Figure 4 show the differences in the internal device architecture between the PowerQUICC II and PowerQUICC II Pro families. Note that the bus architecture shown is a theoretical summary of the PowerQUICC II Pro's capability. For further details regarding the PowerQUICC II family bus architecture, refer to *MPC8260 Dual-Bus Architecture and Performance Considerations* (AN2335). For details on the architecture of a PowerQUICC II Pro device, refer to the device reference manual.

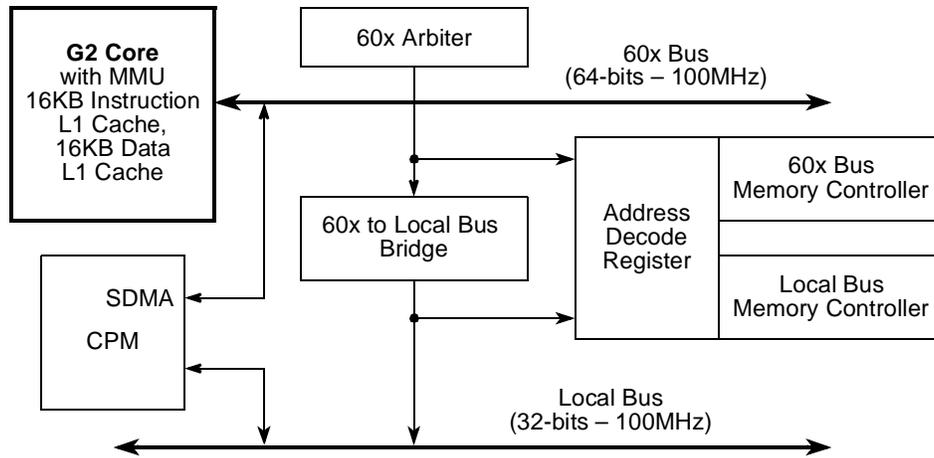


Figure 3. MPC82xx PowerQUICC II Architecture

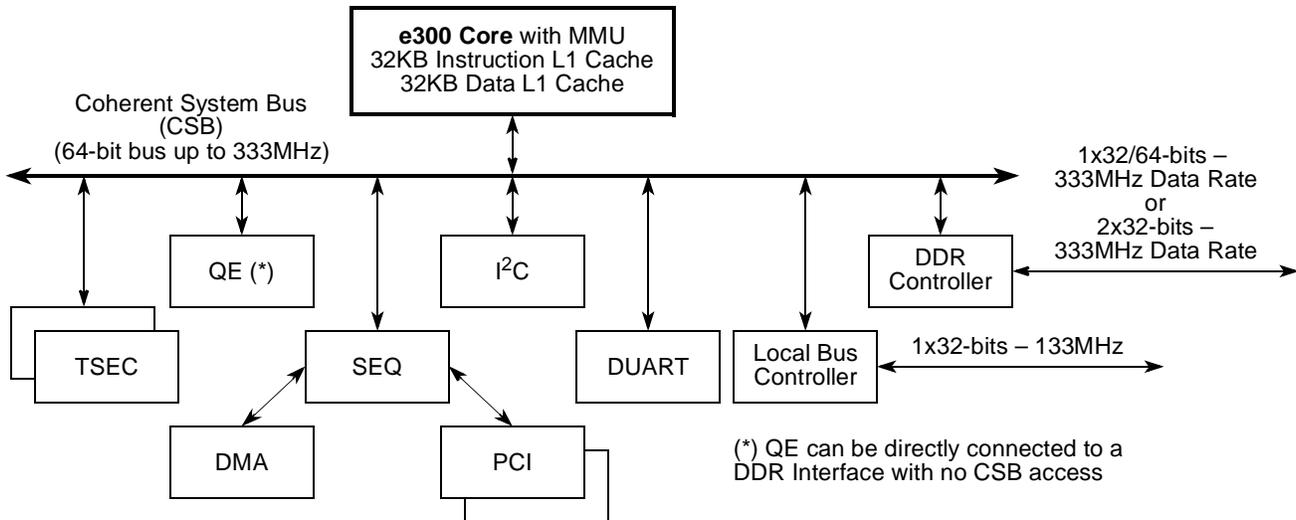


Figure 4. MPC83xx PowerQUICC II Pro Architecture

The PowerQUICC II dual-bus architecture is optimized for CPM accesses to the local bus (direct access) and the G2 core accesses (or another 60x bus master) via the 60x bus. With this configuration, both transactions can be carried out simultaneously. However, this architecture has the following limitations:

- No burst access capability on the local bus for CPU/60x bus master.
- Local bus memory area is not cacheable (due to the lack of burst capability).
- A strong dependency between the CPU/CPM/local bus/60x bus clock frequencies.

As Figure 4 shows, the PowerQUICC II Pro internal bus architecture is built around the coherent system bus (CSB). Within the CSB architecture, the CSB arbiter and bus monitor unit manages transactions between all the possible CSB masters (e300 core, QUICC Engine, TSECs, PCI external masters, DMA ...and so on) and targets (PCI, local bus, DDR controller). The CSB arbiter and bus monitor unit is responsible for:

- Bus transaction arbitration
- Transaction address mapping management
- Transaction coherency management

1.1 Bus Transaction Arbitration

The CSB carries the transactions between most of the possible masters and the targets, except for the QUICC Engine, which has direct access to secondary DDR (MPC835x/6x only). As a shared resource, an arbitration mechanism must manage access to this bus. This arbitration is provided by the CSB arbiter and bus monitor unit. The management unit provides each CSB master with its own set of internal signals to request and to get access to the CSB. [Figure 5](#) shows the CSB master arbitration signals.

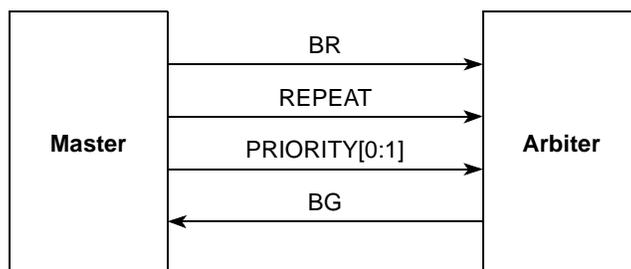


Figure 5. CSB Master Arbitration Signals

Before starting a transaction, a CSB master must be granted the bus. To request bus ownership a master asserts its bus request (BR) signal along with the arbitration signals REPEAT and PRIORITY[0:1]. The bus arbiter later asserts the corresponding bus grant (BG) signal to the requesting master depending on the system state and arbitration mechanism. When BG is asserted to the requesting master, the transaction can begin. The current bus master can transfer 1 to 8 bytes per transaction.

A priority level (0–3) is assigned to all possible bus masters. The arbitration mechanism follows these rules:

- For each priority level, a fair arbitration scheme is used (round-robin scheme).
- For priority levels 1, 2 and 3, a place holder is reserved for lower-level arbitration rings.
- Each master can change its priority level at any time to provide a determinist latency.

[Figure 6](#) shows the CSB mastership algorithm. Some MPC83xx peripherals, such as the TSECs can change their normal priority level to the highest level when moving to emergency mode. For example, if a TSEC receive FIFO reaches a programmable threshold, an emergency mode is automatically triggered so the TSEC can get access to the CSB as quickly as possible, allowing the data to be passed to system memory (DDR, local bus). This avoids an overrun situation. This feature contributes to system stability in a peak load condition. As the TSEC receive FIFO recovers to a normal level (threshold), the TSEC recovers its normal priority level.

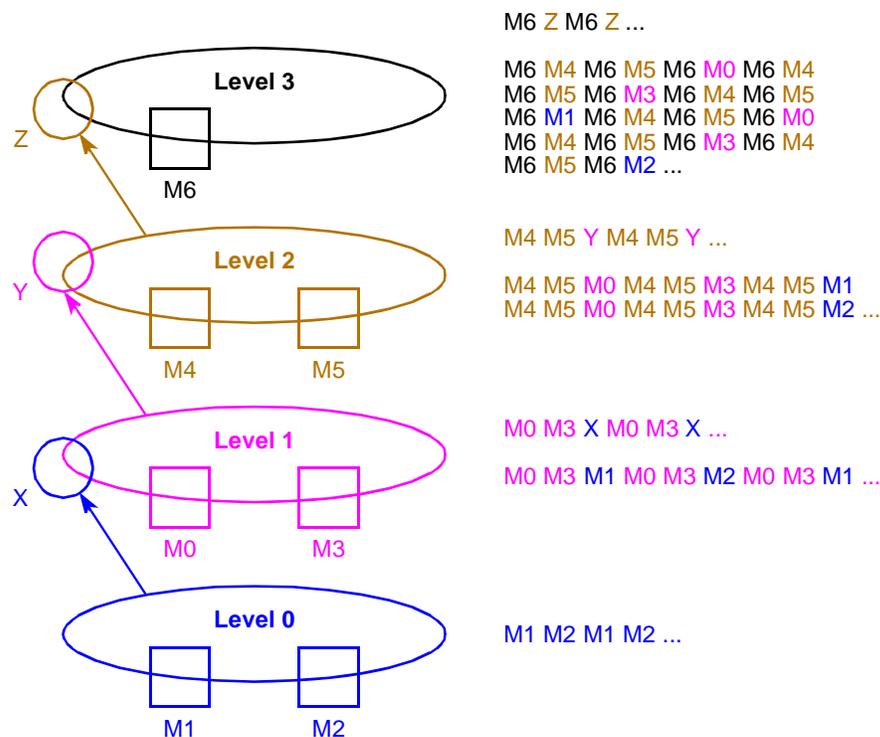


Figure 6. Coherent System Bus Master-Ship Algorithm

1.2 Additional Coherent System Bus (CSB) Features

To ensure a high level of performance from the MPC83xx system architecture, the internal CSB has features discussed in this section.

1.2.1 Address/Data Timeout

To avoid a system deadlock, a programmable time-out for both the address tenure and the data tenure, ensures that any transaction does not last more than the programmed time-out period. If either the address or the data tenure exceeds the address/data time-out, a bus error condition is detected and an interrupt is generated (MCP or regular interrupt).

1.2.2 Repeat Mode

After the first transaction completes, a peripheral can keep ownership of the bus for a programmable number of additional transactions (repeat mode). The number of additional transactions is configured in the arbiter configuration register (ACR). For the MPC83xx, all masters except PCI share the same repeat count capability. The PCI block has a specific repeat count because it can require more successive transactions than other masters.

NOTE

Although up to 8 successive transactions can be programmed for the repeat mode, a maximum of 4 is recommended for a non-PCI master.

1.2.3 Bus Parking

The CSB allows address bus ownership to be tied to a specific master when no other master requests CSB ownership. This mechanism reduces the CSB access latency for the parked master because there is no need for the bus request/bus grant negotiation.

1.3 Transaction Address Mapping Management

All the internal targets of the MPC83xx (like the memory controllers or the PCI interface) are mapped into a 32-bit address space. As [Table 2](#) and [Table 3](#) show, this local memory map is defined by a set of 9 windows for the MPC834x derivatives and 11 for the MPC835x/6x. Each window maps a region of memory to a particular target.

Table 2. Local Access Window Definition for the MPC8349

Window Number	Target Interface	Memory Size
0	Configuration registers	1 Mbyte fixed
1	Local bus	4 Kbytes to 2 Gbytes
2	Local bus	4 Kbytes to 2 Gbytes
3	Local bus	4 Kbytes to 2 Gbytes
4	Local bus	4 Kbytes to 2 Gbytes
5	PCI	4 Kbytes to 2 Gbytes
6	PCI	4 Kbytes to 2 Gbytes
7	DDR SDRAM	4 Kbytes to 2 Gbytes
8	DDR SDRAM	4 Kbytes to 2 Gbytes

Table 3. Local Access Window Definition for the MPC8360

Window Number	Target Interface	Memory Size
0	Configuration registers	2 Mbyte fixed
1	Local bus	4 Kbytes to 2 Gbytes
2	Local bus	4 Kbytes to 2 Gbytes
3	Local bus	4 Kbytes to 2 Gbytes
4	Local bus	4 Kbytes to 2 Gbytes
5	PCI	4 Kbytes to 2 Gbytes
6	PCI	4 Kbytes to 2 Gbytes
7	DDR SDRAM	4 Kbytes to 2 Gbytes
9	DDR SDRAM	4 Kbytes to 2 Gbytes
10	Secondary DDR SDRAM	4 Kbytes to 2 Gbytes
11	Secondary DDR SDRAM	4 Kbytes to 2 Gbytes

1.3.1 Default Boot ROM Region

At power-on PORESET, a default window must enable the CPU to boot and fetch its first instructions. This default window—called the boot ROM region—is configured during the PORESET process by three bits of the reset configuration word high RCWH[ROMLOC]. This feature allows the default 8 Mbyte boot ROM to be located on the local bus (GPCM 8/16 or 32 bit ROM), on a PCI interface, or on a DDR interface.

1.3.2 Configuration Registers

The local window 0 maps a fixed size window to access all memory-mapped configuration, control, and status registers collectively referred to as internal memory-mapped registers. The window size is 1 Mbyte for MPC834x and 2 Mbyte for MPC8358/60 devices. Window 0 is mapped by default to address 0xFF400000 of the internal 32 bit address space. It can be relocated by modifying the IMMRBAR register (IMMRBAR = 0xFF400000 by default).

1.3.3 Local Access Window Configuration Example

Figure 7 shows an MPC8349 configured to define its boot ROM location on the local bus, using a GPCM with an 8-bit bus width setting. As a result, the default local memory map of the MPC8349 is used at PORESET.

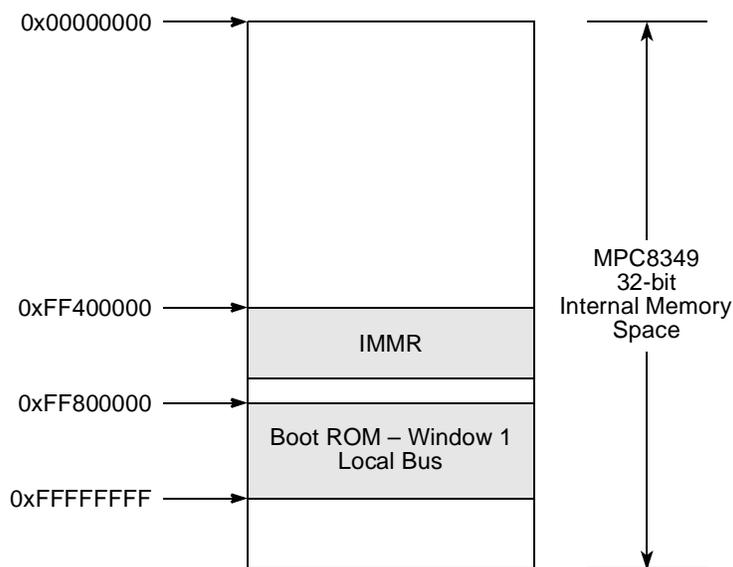


Figure 7. MPC8349 Default Local Bus Memory Map Window

NOTE

In this example, the boot ROM is configured through the reset configuration word to the region 0xFF800000–0xFFFFFFFF.

After configuration, the MPC8349 32-bit internal memory space can be defined to support one PCI interface (256 Mbytes), one DDR interface (2 Gbytes), and one local bus interface (256 Mbytes). See Figure 8. In this configuration, address windows 3, 4 and 6 are not used.

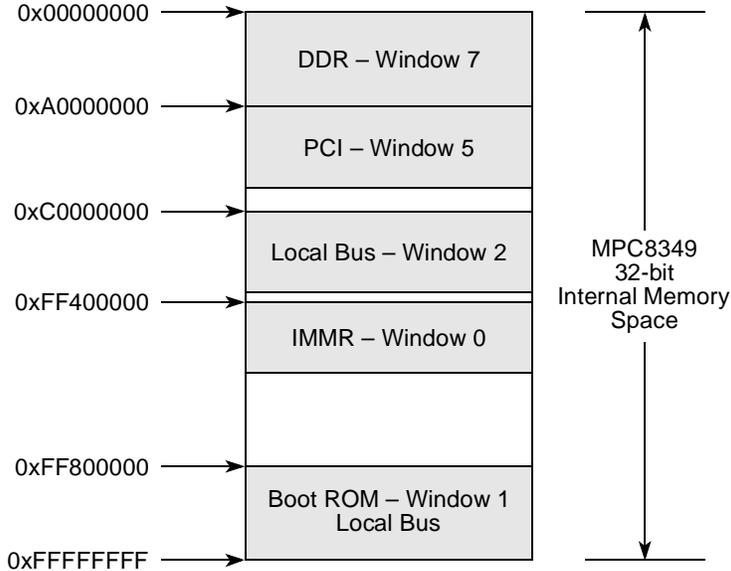


Figure 8. MPC8349 Memory Map Windows After Configuration

1.3.4 Direct Access of the QUICC Engine to DDR Memory Interface (MPC835x/6x Only)

On the MPC835x/6x, two additional windows called the QUICC Engine secondary bus access windows are available. The QUICC Engine (QE) has a dedicated local bus called the QUICC Engine secondary local bus to provide it with direct access to both the local bus memory controller and the secondary DDR controller without requesting the CSB arbiter and bus monitor unit for any access to the CSB. As with the normal local access windows of the CSB bus, the two QE secondary bus access windows—one for a direct QE access to the local bus and one for a direct QE access to the secondary DDR interface—support a memory map region of 4 Kbytes to 2 Gbytes.

These additional local access windows enable the e300 core to access the primary DDR interface as the QE can also gain access either to the local bus or the secondary DDR interface. Figure 9 shows an example of the usage of the QE secondary access windows to optimize the processing of both the control path and the data path in a DSLAM line card application.

If the secondary DDR interface is mapped as the QE secondary access window—that is, the QE has a direct access to the secondary DDR interface—the e300 core can still access to this secondary DDR interface via the local access windows 10 and 11 (refer to Table 3) if these windows are configured. However, the e300 and the QE cannot access the secondary DDR interface at the same time.

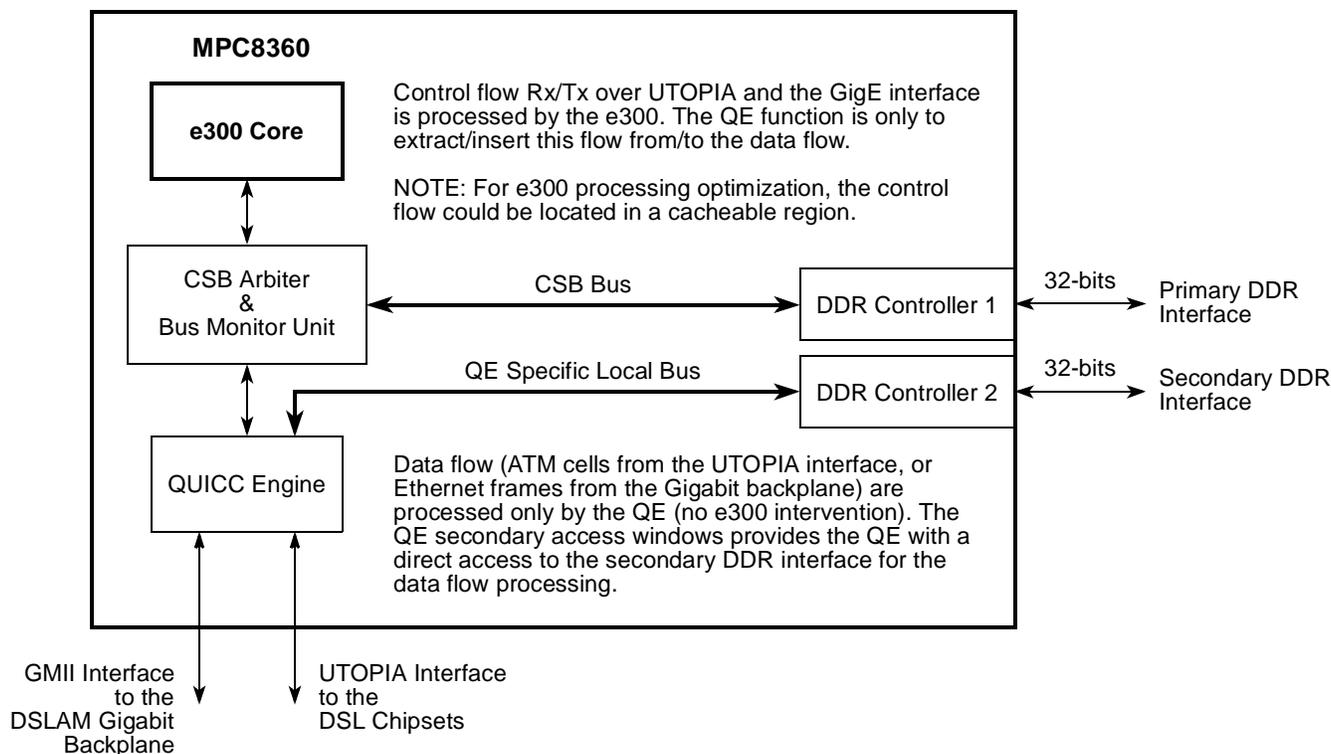


Figure 9. Example Usage of the QUICC Engine Secondary DDR Interface

With this configuration, the QUICC Engine can process the data flow on the secondary DDR interface, while the e300 core runs the application software on the primary DDR interface and manages the control flow.

1.4 Transaction Coherency Management

The CSB arbiter and bus monitor unit also manages the coherency between the data stored in the e300 L1 cache and the data stored in the system memory (DDR interface or local bus). To ensure cache and data coherency, any global transaction carried out by a CSB internal master is snooped. As a result, if such a global transaction targets data already stored in the L1 cache, the following occurs:

- Global write transaction—The data is written to the targeted main memory (DDR or local bus) and to the appropriate L1 cache line (cache update).
- Global read transaction:—For example, if a PCI master reads (global transaction) data located both in DDR and in the e300 L1 data cache, and if the data in DDR is not coherent with the data in L1 cache, the following occurs:
 - SEQ access to the CSB bus to be retried (note PCI master transaction to be retried also if timeout expires)
 - DDR data is refreshed with the data in e300 L1 cache
 - SEQ read transaction on CSB is retried to the DDR, to read the up-to-date data

On the MPC8360/58, any transaction initiated by the QE through the QE secondary bus access windows cannot be snooped because these window accesses are not monitored by the coherency module.

Figure 10 and Figure 11 show an example of data flow in the MPC8349, the reception of Ethernet frames by the TSEC, and their storage in the DDR memory.

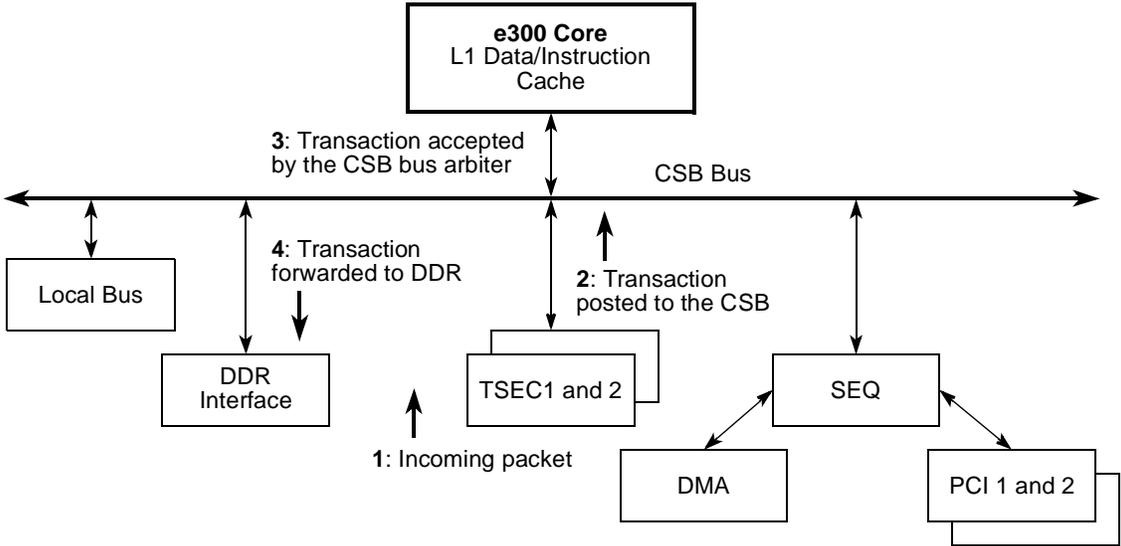


Figure 10. Non-Global Transaction Example (Received Ethernet Frames are Not Snooped)

In this example, the TSEC Rx FIFO reaches the fill threshold, and then the TSEC then posts a request to the CSB arbiter and bus monitor unit to forward the received data to the DDR interface (assertion of the BR signals, the priority signals, and the repeat signal). After acceptance of this request (CSB arbiter and bus monitor unit asserts the internal TSEC BG signal), the TSEC acquires the ownership of the CSB bus and starts with the address tenure of the transaction. The destination address is decoded through the local access windows to identify the target (DDR in this example). The data tenure can then be forwarded and the data placed on the CSB bus and captured by the target (DDR interface).

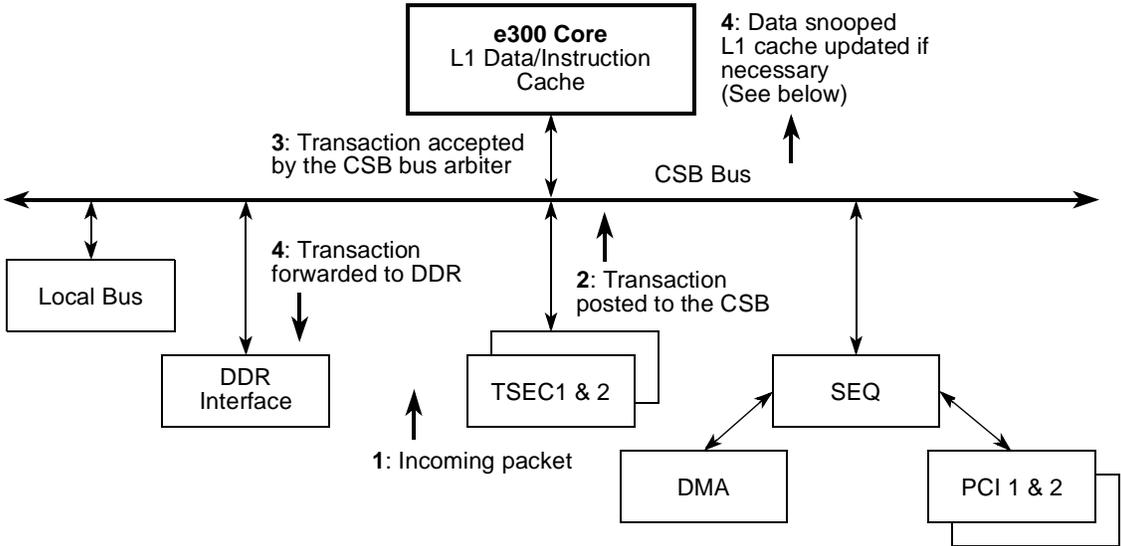


Figure 11. Global Transaction Example (Received Ethernet Frames are Snooped)

Figure 11 shows an example similar to that of Figure 10. However, while posting its transaction request to the CSB arbiter and bus monitor unit, the TSEC also activates the internal global signal to specify that this transaction should be snooped by the CSB arbiter and bus monitor unit when forwarded. As a result, the destination address of the transaction is snooped to check whether the data is in the L1 cache. If so, the data is stored over the DDR interface and the L1 cache is updated to maintain cache coherency.

2 PORESET

The PowerQUICC II Pro PORESET architecture provides a rich set of options that enable, among other configurations, the following:

- e300 core boot from FLASH
- e300 core boot from DDR/PCI
- MPC83xx device configuration from external PCI host/I2C
- e300 core disable (MPC83xx used PCI slave)

As Figure 12 shows, the PORESET sequence process consists of four steps.

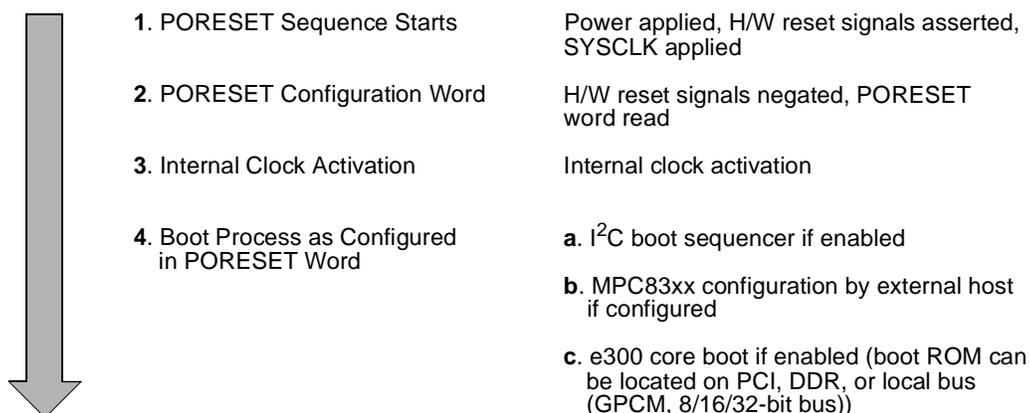


Figure 12. MPC83xx PORESET Sequence

During the second step, four reset configuration signals (device pins multiplexed with other functions when the device is not in a reset state) are sampled to determine which reset configuration word should be used and its location. These pins are listed below:

- CFG_RESET_SOURCE [0:2]—These three pins configure how the PowerQUICC II Pro gets its reset configuration word. The options are as follows:
 - A default hard coded reset configuration word (4 possible options)
 - A reset configuration word loaded from the local bus EEPROM
 - A reset configuration word loaded from an I2C EEPROM (with two sub option depending on the PCI_CLK/PCI_SYNC_IN clock value)

Note the pin assignment description:

- CFG_RESET_SOURCE 0 is multiplexed with LSDA10/LGPL0
- CFG_RESET_SOURCE 1 is multiplexed with LSDWE/LGPL1
- CFG_RESET_SOURCE 2 is multiplexed with LSDCAS/LGPL3

- CFG_CLKIN_DIV—This signal configures the relationship between CLKIN and PCI_SYNC_OUT when the MPC83xx is configured as a PCI host. Note that when the MPC83xx is configured as a PCI agent, this pin must be 0.
CFG_CLKIN_DIV is multiplexed with LGPL5.

Whatever its source (refer to the CFG_RESET_SOURCE [0:2] signal description), a 64-bit reset configuration word configures the device at PORESET and HRESET. This word configures the following:

- MPC83xx internal clock circuitry
 - Local bus memory controller clock mode
 - DDR SDRAM memory controller clock mode
 - System PLL multiplication factor
 - e300 core PLL configuration
 - QUICC Engine PLL division factor (MPC835x/6x only)
 - QUICC Engine PLL multiplication factor (MPC835x/6x only)
- MPC83xx PCI interface
 - PCI host/agent mode
 - PCI 32/64 bit mode (MPC8349 only)
 - PCI 1 internal/external arbiter
 - PCI 2 internal/external arbiter (MPC8349 only)
 - PCI output drive (MPC835x/6x only)
- e300 core activation
 - e300 core enabled/disabled
- Boot parameters
 - Boot memory space
 - Exception vectors: boot ROM(0x00000000 to 0x007FFFFFFF or 0xFF800000 to 0xFFFFFFFF)
 - e300 core: boot address (0x00000100 or 0xFFF00100)
 - Boot sequencer configuration (I2C boot sequencer activation/deactivation)
 - Boot ROM location (PCI, DDR, local bus (GPCM 8/16/32-bit bus))
- TSEC
 - TSEC1 mode (GMII/TBI/RGMII/RTBI) (MPC834x only)
 - TSEC2 mode (GMII/TBI/RGMII/RTBI) (MPC834x only)
- Miscellaneous
 - Big/little endian mode
 - Watchdog timer enable/disable
 - Secondary DDR IO enable/disable (MPC835x/6x only)

During the fourth and final step of the PORESET sequence, there are three complementary ways to configure the MPC83xx internal memory-mapped registers, as summarized in [Figure 13](#).

2.1 I²C Sequencer

If activated, the I²C sequencer can load the initialization sequence is generated to configure any of the MPC83xx internal or external registers in any memory-mapped address. This process can be used, for example, to pre-program the e300 core to boot directly from memory devices such as burstable FLASH or DDR memories. The I²C sequencer accesses the entire MPC83xx internal 32-bit memory space 1 Mbyte at a time, via a specific window. This access is carried out in two steps:

- The alternate configuration space (ACS) in the EEPROM data format bit is set (refer to section 4.4.3.2.3 in the *MPC8349E Family Reference Manual*), supplying the desired registers address which is pointed to by the alternate configuration base address register (ALTCBAR).
- By combining the base address in the ALTCBAR with that of the 20 bits of address offset supplied from the serial ROM to generate a 32-bit address that is mapped to the target specified in ALTCAR

2.2 PCI Master

When configured as a PCI slave, the MPC83xx architecture allows a PCI host to gain access to the MPC83xx internal configuration and status registers. This allows an external PCI host to control and configure the MPC83xx. At this stage the e300 core has still not booted and therefore does not run any code.

2.3 e300 Core

If enabled, the e300 core boots from the boot ROM as configured in the reset configuration word high (RCWH). If the e300 core is disabled using the RCWH[COREDIS] bit, the e300 cannot fetch any boot code. The MPC83xx device is then used as a slave platform by an external master on the PCI. The PCI master can gain access to any peripheral of the PowerQUICC II Pro platform such as the QUICC Engine, TSECs, DDR, or DMA controllers.

The fourth step of the MPC83xx PORESET sequence provides complementary ways to configure the device. [Figure 13](#) summarizes these three possibilities to configure the MPC83xx internal memory-mapped registers.

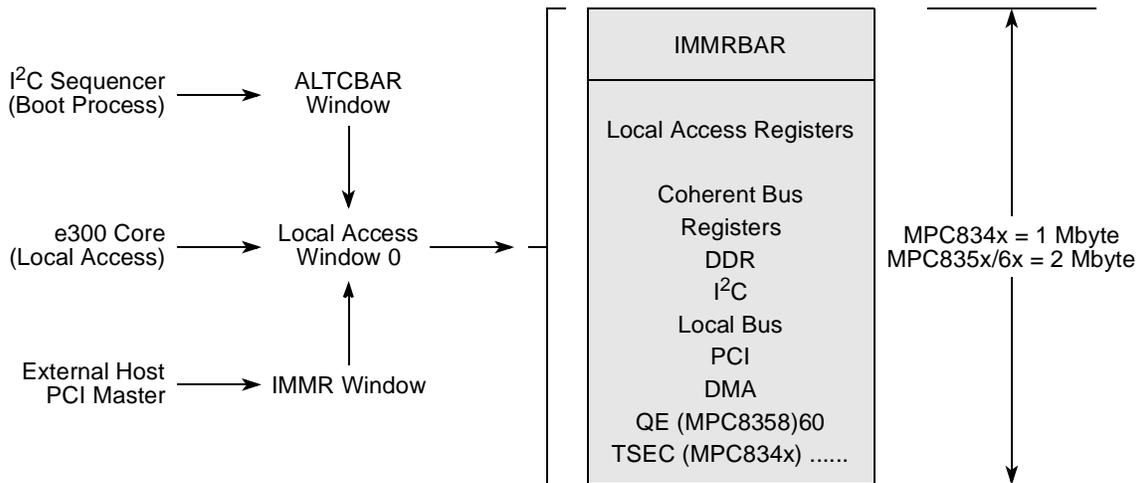


Figure 13. MPC83xx Internal Configuration Masters

NOTE

When the MPC83xx is active, both the e300 core (if enabled) or an external PCI master still access (both read and write access) the internal memory mapped registers.

3 DMA Controller

The MPC83xx DMA controller is a dedicated 4-channel, independent, general-purpose controller that can be used by both local and remote masters to transfer data between any memory-mapped area of the device (for example, PCI, local bus, or DDR). The DMA controller is similar to that of the IDMA on the MPC82xx family, with traditional direct/auto buffer and chaining modes. The chaining mode allows the user to set up a series of buffer descriptors referenced from a linked list. The DMA controller can then walk through multiple buffer descriptors, performing complex DMA transactions.

4 e300 Core

This section describes the core differences and improvements when moving from a PowerQUICC II family device to a PowerQUICC II Pro device. The e300 core is an enhancement of the G2 and G2LE (MPC827x/MPC8280 only) PowerPC cores available on all current PowerQUICC II devices. The e300 core is fully backward-compatible with the G2 and G2LE cores and implements enhancements and extensions over these cores, improving performance through the following:

- Cache size
- MMU size
- MMU extensions
- Debug feature
- Performance

For details on the enhancements and extensions that apply to the e300 core over the G2 and G2LE cores refer to the *MPC8349E PowerQUICC II Pro Integrated Host Processor Family Reference Manual* and the *e300 PowerPC Core Reference Manual*. The most noticeable differences between the two processors include:

- e300 cache increased from 16 Kbyte instruction + 16 Kbyte data (G2/G2LE) to 32 Kbyte instruction + 32 Kbyte data
 - G2 4-way set associative, LRU algorithm improved to e300 8-way set associative PLRU algorithm
 - G2 1-3 way or entire cache way locking improved to e300 7 of 8 way or entire cache way locking
 - No cache parity support on G2 improved to e300 instruction and data cache parity support
 - e300 instruction cancel and burst fetch extensions
- G2 (not including G2LE) 4x IBAT + 4x DBAT entries improved to e300 8x IBAT + 8x DBAT entries
- G2 support for one-level address pipelining improved to e300 support for one-and-a-half level address pipelining

- e300 additional core debug register fields
- e300 additional core power saving modes

To port software across cores, the e300 is backward-compatible with the G2/G2LE PowerPC cores. Improvements can be gained from reworking (recompiling) original G2/G2LE code to use the enhancements and extensions on the e300 core as well as the enhanced debug features. The following sections describe these benefits.

4.1 Cache Enhancements

The e300 has an increased cache—double the size of the G2/G2LE (from 16 Kbyte instruction + 16 Kbyte data for the G2LE to 32 Kbyte instruction + 32 Kbyte data for the e300 core). This larger cache immediately provides speed and code efficiency benefits. The e300 core includes features such as full cache parity support, instruction cancel, and burst fetch extensions that give it a performance advantage over its predecessor. The extended cache size on the e300 core is relatively transparent to the user. However, with parity checking on both the instruction and data cache, they can now act as a source for a machine check (or checkstop depending upon MSR[ME]) exception (exception vector 0x00200). Ensure that software can distinguish between the different parity error sources.

The cache parity checking functionality is enabled through HID0[ECPE] (enable cache parity errors). Cache parity errors are reported through SRR1[10]. For details on the HIDx and SRRx registers, refer to the device reference manual.

4.1.1 Instruction Cancel (HID2.ICCANEN Bit)

The e300 cache instructions allow improved use of the instruction cache during cancel operations. Instructions are routinely fetched ahead of time in the e300 super-scalar architecture but then may be cancelled due to instruction redirection (for example, branches or exceptions). The new extensions allow:

- Instruction fetches issued to the bus unit to be cancelled if they have not yet started on the bus for the first time
- New instruction hits to occur in the instruction cache while a cancelled instruction fetch is currently busy on the bus
- A second instruction fetch to be issued to the bus unit while a cancelled instruction fetch is currently busy in the bus unit
- A new instruction fetch to be issued to the bus unit while the remaining instructions of a burst-fetch are being returned from the bus

To illustrate the advantage of these e300 core enhancements, consider the following instruction sequence, where instruction D is a branch instruction:

1. A
2. B
3. C
4. D
5. E

The e300 branch unit predicts on this instruction that the next instruction fetched is P then Q, R, S (which are then fetched into the instruction queue). However, on completion of instruction D, we find that the branch was incorrectly predicted and E should have been the next instruction. We can address this situation as follows:

1. The e300 flushes the instruction queue (P, Q, R, S). This also happens with the G2 architecture.
2. The e300 instruction fetches issued to the bus unit are cancelled if they have not yet started on the bus for the first time. The current instruction fetch is for T. The instruction fetch has not generated the instruction to be loaded at this stage. The result on the e300 architecture is the cancellation of the fetch generated by instruction T. The G2 architecture cannot cancel this fetch and must carry it through, giving the e300 a performance advantage.
3. New instruction hits occur in the instruction cache while a cancelled instruction fetch is currently busy on the bus. While the T instruction is under cancellation, the E instruction fetch can occur and generate a cache hit within the e300 architecture. However, the G2 core cannot perform an instruction fetch from instruction cache until the transaction is cancelled on the bus.
4. Issue a second instruction fetch to the bus unit while a cancelled instruction fetch is currently busy in the bus unit. This is the same as option 3 except that the e300 architecture instruction fetch can generate a fetch after a miss or non-cacheable access (that is, the instruction fetched is not in the cache and must go external) however, the G2 architecture does not support this.
5. A new instruction fetch is issued to the bus unit while the remaining instructions of a burst fetch are being returned from the bus. T has generated a burst request. With the e300 core, E can be fetched while the T fetch (burst) is cancelled. Once again, this is not possible with the G2 core.

4.1.2 Burst Fetch

e300 cache performance is also enhanced over G2/G2LE through burst fetching of instructions from cache-inhibited instruction space (as for cacheable instruction space). This feature improves performance during accesses to instruction space designated as cache prohibited. With this new performance feature, up to an entire cache line may be returned (up to 8 instructions) with one bus operation. Remaining instructions in the burst read are normally forwarded to the instruction fetch unit as would normally be requested. The burst fetch feature is enabled through HID2[IFEB] (Instruction Fetch Burst Extension). For details on to HIDx and SRRx registers, refer to the device reference manual.

4.2 MMU Enhancement

The e300 core has further enhancements to the MMU over the G2/G2LE core. Highlights of these enhancements are discussed here, with information on code/setting changes required by user software. The e300 core MMU (memory management unit) has an additional four instruction BAT registers (IBATs) and data BAT registers (DBATs). This allows effective hardware address translation of 256 Mbytes from 32-bit effective to physical. However, this increase exists only over the G2 (MPC82xx) processors. The additional BAT registers are enabled/disabled through HID2[HBE] (high BAT enable). For details on the HIDx and SRRx registers, refer to the device reference manual.

4.3 Address Pipelining

The G2/G2LE core can pipeline the address bus to one level, so the core can support up to two outstanding bus transactions at any time or two pipeline slots on the bus (one additional address). A new bus transaction cannot start until a pipeline slot is available. A pipeline slot does not become available until a previous slot completely finishes on the bus (address and data complete). The e300 pipeline performance extension allows a new pipeline slot to become available when a previous transaction is granted the data bus. This is referred to as “one-and-a-half-level” pipelining. This extension provides greater bus use within the e300-based system.

4.4 Power Saving

The e300 core adds a nap mode to the doze and sleep modes on the G2/G2LE cores. Nap mode can put the e300 core into shutdown while both the PLL and time base counter remain active. The doze function on either of the cores allows only the PLL, time base, and snooping to remain active. In contrast, in sleep or stop mode, all core functionality stops so hardware can stop the PLL. The e300 NAP functionality is enabled through HID2[NAP]. For details on the HIDx and SRRx registers, refer to the device reference manual.

5 Power Enhancements

The MPC83xx is an SoC (system-on-chip) architecture with high performance but low power. This section describes the requirements for the MPC83xx PowerQUICC II Pro in relation to the MPC82xx PowerQUICC II.

The strict power-up sequencing required with the MPC82xx is not a constraint with the MPC834x. It is not necessary for the MPC834x core supply (Vdd) or I/O supplies (GVdd, LVdd1+2, OVdd) to be powered up in any particular sequence. The MPC82xx requires that the core supply and I/O supplies be brought up with strict tolerance; for details, see the device hardware specifications document.

However, during power ramp-up to the MPC834x device, the I/O pins from the device can be actively driven. Therefore, PORESET should be asserted to the MPC834x before power is supplied to the device to ensure that I/O pins are tri-stated for a minimum time and prevented from adversely affecting surrounding logic. For details, see the device hardware specifications document.

6 Debug

The PowerQUICC II Pro family has several features to help in debugging designs at the board implementation stage. This section highlights new debug features gained in moving from the MPC82xx to the MPC83xx.

Debug features of the MPC83xx are as follows:

- JTAG/COP Interface
- e300 enhanced debug functionality
- DDR debug capability
- Local bus debug capability

The JTAG/COP interface is a standard also found on the MPC82xx family of devices. However, there are differences between the MPC82xx and MPC83xx core debug facilities.

6.1 e300 Core Debug Register Additions

There are four new core debug registers on the e300 core that are unavailable on the G2/G2LE cores:

- Instruction address breakpoint control register (IBCR)
- Data breakpoint registers (DABR, DABR2)
- Data address breakpoint control register (DBCR)

The IBCR controls the compare and match type conditions for the instruction address breakpoint register (IABR) and IABR2. The control register allows the user to break on a wide combination of instruction address conditions—for example, if the current instruction address is greater than or equal to, equal to, less than one value of two, and so on. This allows the user great flexibility when debugging e300 core instruction accesses. For full details on the e300 implementation of the IABRs and IBCRs, refer to their description in the *MPC8349E PowerQUICC II Pro Integrated Host Processor Family Reference Manual*.

The DABR and DBCR registers are used to the same effect as the IABR and IBCR. However, these registers break code operation upon data accesses to a double word address. The address compare functions include those discussed in the previous section for instruction address breakpoints. For full details on the e300 implementation of the DABR and DBCR, refer to their description in the *MPC8349E PowerQUICC II Pro Integrated Host Processor Family Reference Manual*.

6.2 Memory Debug Support

The MPC834x family also supports debug capabilities across the two main bus interfaces on the device, namely the DDR SDRAM bus interface and the local bus interface. The interfaces provide debug information to the user by driving the device pins MSRCID[0:4], MDVAL and LSRCID[0:4], LDVAL respectively. This information can then be used to identify the internal source of a transaction which reached the DDR SDRAM or local bus interfaces.

Enabling the DDR debug configuration allows the DDR SDRAM source ID field and data valid strobe to be driven onto one of two sets of pins.

1. ECC pins (ECC checking and generation are disabled in this case)
2. UART pins (UART operation is disabled in this case)

Whichever set of pins is chosen to retrieve the DDR interface debug information from the affected pins must be disconnected from any on-board devices that may drive them. For example, if ECC signals are to be used, signals driven from the SDRAMs must be electrically disconnected from the ECC I/O pins. Equally, if UART signals are to be used, signals driven by UART devices must be electrically disconnected from the UART I/O pins on the MPC83xx. For full details on the MPC834x implementation of the DDR bus debug configuration, refer to the *MPC8349E PowerQUICC II Pro Integrated Host Processor Family Reference Manual*.

The local bus debug configuration operates in the same way as described for the DDR debug configuration. The local bus debug configuration enables the SDRAM source ID field and data valid strobe to be driven onto the MPC8345 UART pins. As with the DDR bus debug configuration, UART operation is disabled when this operational mode is enabled for the local bus, and any signals driven by UART devices must be

electrically disconnected from the UART I/O pins. The MPC83xx DDR memory controller also offers ECC error injection support to aid in rapid system debug. This feature can be used to inject single bit errors or a mirror copy of the most significant byte of data to the into the ECC memory. For details, refer to the *MPC8349E PowerQUICC II Pro Integrated Host Processor Family Reference Manual*.

7 Local Bus Controller

The MPC834x local bus controller performs similar functions to that of the MPC82xx family. The three main MPC82xx memory controller machines are supported on the MPC834x devices:

- GPCM—General-purpose chip-select machine for classic memory interfaces such as SRAM memory or FLASH
- UPM—User-programmable machine that configures an internal timing generator to create non-standard timings for peripheral devices
- SDRAM—Single data rate SDRAM controller allowing SDRAM devices to be gluelessly connected

Although functionally similar, the MPC834x implementation of the UPM has three dedicated UPMs unique to the local bus. With the MPC82xx, these UPMs are also shared with the 60x bus. Similarly, with the MPC82xx, the twelve available chip selects are shared with the 60x bus. However, the MPC834x has four dedicated chip selects specific to the local bus. Also, it removes the support for bank-based interleaving on the SDRAM controller available on the MPC82xx. The MPC834x SDRAM controller supports only page-based interleaving because page-based interleaving allows a greater degree of flexibility and control when accessing connected SDRAM devices.

The major difference between the two families of devices is that the local bus controller on the MPC83xx can operate up to 133 MHz, but the MPC82xx can operate only up to 100 MHz. The MPC83xx local bus runs with a fixed frequency referenced from the main system frequency —SYSCLK. Note that the local bus controller runs synchronized with the CSB clock. The MPC83xx local bus programming model is the same as that of the MPC82xx (although only four chip-selects are supported for the MPC834x and eight for the MPC835x/6x). Each individual chip-select is configurable to 8, 16, or 32 bits wide. Each individual chip-select supports up to 2 Gbytes of address space. The local bus supports both natural parity and read-modify-write (RMW) data parity checking.

To save package pin count, the local bus on PowerQUICC II Pro devices has a 32-bit multiplexed address/data interface. This represents a significant change when moving from the MPC82xx to the MPC83xx because any MPC83xx local bus design requires an external buffer and latch combination with which to de-multiplex the multiplexed address/data signals. The logic for controlling these external devices is provided through the MPC83xx local bus controller (LALE, LBCTL and /LCS).

8 Double Data Rate SDRAM Memory Controller

A major new feature of the MPC83xx is a dual data rate (DDR) memory controller, as described in the following documents available at the web site listed on the back cover of this document:

- *MPC8349E PowerQUICC II Pro Integrated Host Processor Family Reference Manual*
- *Programming the PowerQUICC III DDR SDRAM Controller (AN2583)*
- *Hardware and Layout Design Considerations for DDR Memory Interfaces (AN2582)*

The MPC83xx family (DDR) SDRAM currently supports DDR Type 1 SDRAM up to 333 MHz. Because DDR memory is clocked on both edges of the input clock, the actual maximum clock rate is 167 MHz. The MCK# [5:0] signals are simply the inverted clocks from the MCK [5:0] signals and not true differential clock signals.

The MPC82xx 60x bus allows users to connect to SDRAM, Flash memory, and other 60x bus-compatible peripherals. However, the MPC834x DDR controller is a dedicated 64-bit data bus (+ 8-bit ECC) with no external master capability, and it is used exclusively for DDR memories as the main system bus. Note that connection to other external masters can be readily implemented through PCI. The DDR controller supports discrete or DIMM DDR memories between 64 Mbytes and 1 Gbyte. The DDR controller has four chip selects, each used to access a (theoretical) maximum of 1 Gbyte addressing space. The full 4 Gbytes cannot be used for DDR because 4 Gbytes is the maximum addressing that the MPC834x can handle. Thus, the DDR controller is restricted to a maximum addressing limit of 3.5 Gbytes to allow space for other connected peripherals in the memory map.

8.1 Source Synchronous Operation

The DDR controller supports source synchronous clock control, with data captured at the memory interface and at the controller interface using the data strobe rather than the clock. Data can be captured on both the falling and rising edges of the strobe, doubling the speed of the data bus over standard SDRAM interfaces. However, with the enhanced technology in the MPC83xx DDR controller, extra care is required to ensure that the design is stable. To understand MPC83xx DDR design methodology, review documents such as *Hardware and Layout Design Considerations for DDR Memory Interfaces* (AN2582).

8.2 Clocking

With traditional clock design, no more than three input pins should be supplied with one clock because this can cause the resulting clock to be degraded under loaded conditions. Therefore, the DDR controller supplies six clock pairs for use with DDR memory. Some external clock buffering is additionally required if users decide to deviate beyond this clock design guideline.

8.2.1 Stub Series Terminated Logic (SSTL-2)

The Stub Series Terminated Logic 2.5V (SSTL-2) interface is the leading industry choice for next-generation technology in high-speed memory subsystems. The MPC834x DDR interface supports this technology. SSTL is optimized for the main memory environment, which has long stubs off the bus due to the DIMM routing traces. [Figure 14](#) shows the most common technique for single-ended termination. Manufacturer application notes strongly recommend single-ended termination because adjusting the R_S and R_T values easily allows adaptation of the waveform shape and amplitude which, in turn, can improve signal integrity and noise margin. However, this style of circuit burns static power.

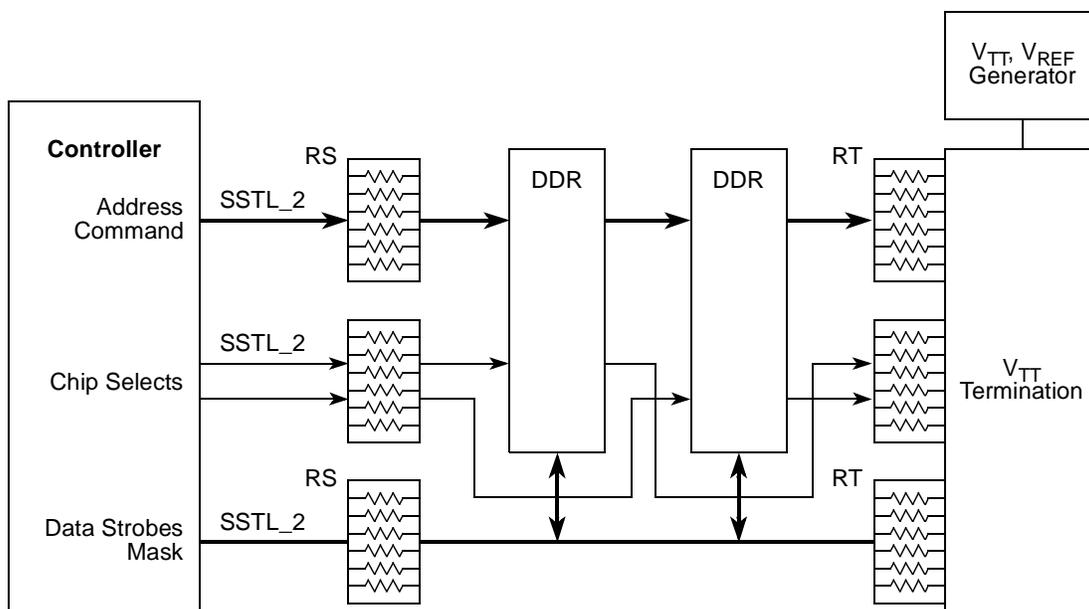


Figure 14. Single-Ended Termination Technique

8.2.2 Sleep Mode Support for Self Refreshing SDRAM

The DDR memory controller supports sleep mode for self-refreshing SDRAM. If clocks or power must be withheld from the MPC834x during operation, this feature can be used prior to shutting down the processor. The SDRAM self-refresh mode is enabled through `DDR_SDRAM_CFG[SREN]` (self refresh enable). If enabled, the DDR memory controller places the DDR SDRAM into self-refresh mode before shutting down clocks and signalling to the e300 core. Self-refresh mode guarantees that the memory content is valid while the memory controller and its clocks are off.

8.2.3 Dynamic Power Management

Dynamic power management allows users to maximize the bandwidth capabilities of the DDR SDRAM by using on-the-fly power management. As the DDR transaction pipeline begins to empty, the device automatically uses the clock enable to put the device into power conservation mode. As new DDR transactions arrive, the device comes out of power conservation mode and begins pipelining those new transactions for execution.

9 PCI

The MPC83xx family PCI block is functionally similar to that on the MPC82xx. However, there are some key additions:

- MPC8349 support for 32-bit/64-bit PCI
- MPC83xx increased to 5x request/grant signal pairs¹ (over MPC82xx 3x Request/Grant signal pairs)

1. PCI1 supports 5x request/grant pairs. PCI2 (MPC8349 only) supports 3x request/grant pairs.

The MPC83xx family supports both single (MPC8343/8347) and dual (MPC8349) PCI modules. The MPC83xx PCI module is version 2.2 compliant, supporting transaction speeds between 16 to 66 MHz. As with the MPC82xx device, the MPC83xx PCI controller can either use its internal arbiter to control PCI transactions or an external arbiter. The MPC83xx PCI interface can be configured to operate either as an agent or host, or it can be configured as a master or slave PCI device. The PCI controller can be configured from the PCI bus while it is in agent mode. However, with the dual PCI (MPC8349 only), one PCI module (PCI1) is available when the device is configured in agent mode. With the MPC8349 configured in host mode, the two PCI interfaces are available autonomously.

An address translation mechanism maps PCI memory windows between the PCI bus and the internal bus. As a master device, the MPC834x PCI controller manages both memory and I/O transactions, as a slave device, it controls only memory transactions.

The MPC83xx supports five separate request/grant pairs to support up to five external PCI masters. The PCI interface is a dedicated interface and does not multiplex its pins with the local bus as on the MPC82xx. On the MPC82xx, the local bus pins and PCI are multiplexed, and one of the two interfaces can be used. This restriction does not exist on the MPC834x. However, the PCI interface is multiplexed with some GPIO pins, and dual-PCI support with the MPC8349 supports either 2x 32-bit PCI ports or 1x 64-bit PCI slot. Combinations of 64-bit and 32-bit ports are not possible due to the MPC8349 package pin multiplexing.

10 Integrated Programmable Interrupt Controller (IPIC)

The MPC83xx family supports an integrated programmable interrupt controller (IPIC) that supports 8 external and 35 internal discrete vectorized interrupt sources and has full functional and programming compatibility with the MPC82xx interrupt controller. Therefore, the migration from the MPC82xx interrupt controller to the MPC83xx family is simplified.

Figure 15 shows the MPC83xx IPIC structure. There are some enhancements to the MPC83xx IPIC over the MPC82xx interrupt implementation. In addition, to MPC82xx INT and MCP, new interrupt types are added to the MPC83xx family:

- System Management Interrupt ($\overline{\text{SMI}}$)
- Critical Interrupt ($\overline{\text{CINT}}$)

In the MPC834x IPIC, each interrupt can be programmed into a subgroup with up to eight interrupts with programmable priority. The highest two interrupts of each subgroup can be programmed to assert $\overline{\text{INT}}$, $\overline{\text{SMI}}$, or $\overline{\text{CINT}}$ interrupts. The lower six interrupts may assert only the $\overline{\text{INT}}$ signal. This functionality is programmed through the system external interrupt control register (SECNR). At $\overline{\text{SMI}}$ or $\overline{\text{CINT}}$ assertion, the systems interrupt service must determine the source of interrupt by reading SMVCR and SCVCR registers, respectively. SIVCR is common to both devices and used to determine the $\overline{\text{INT}}$ source.

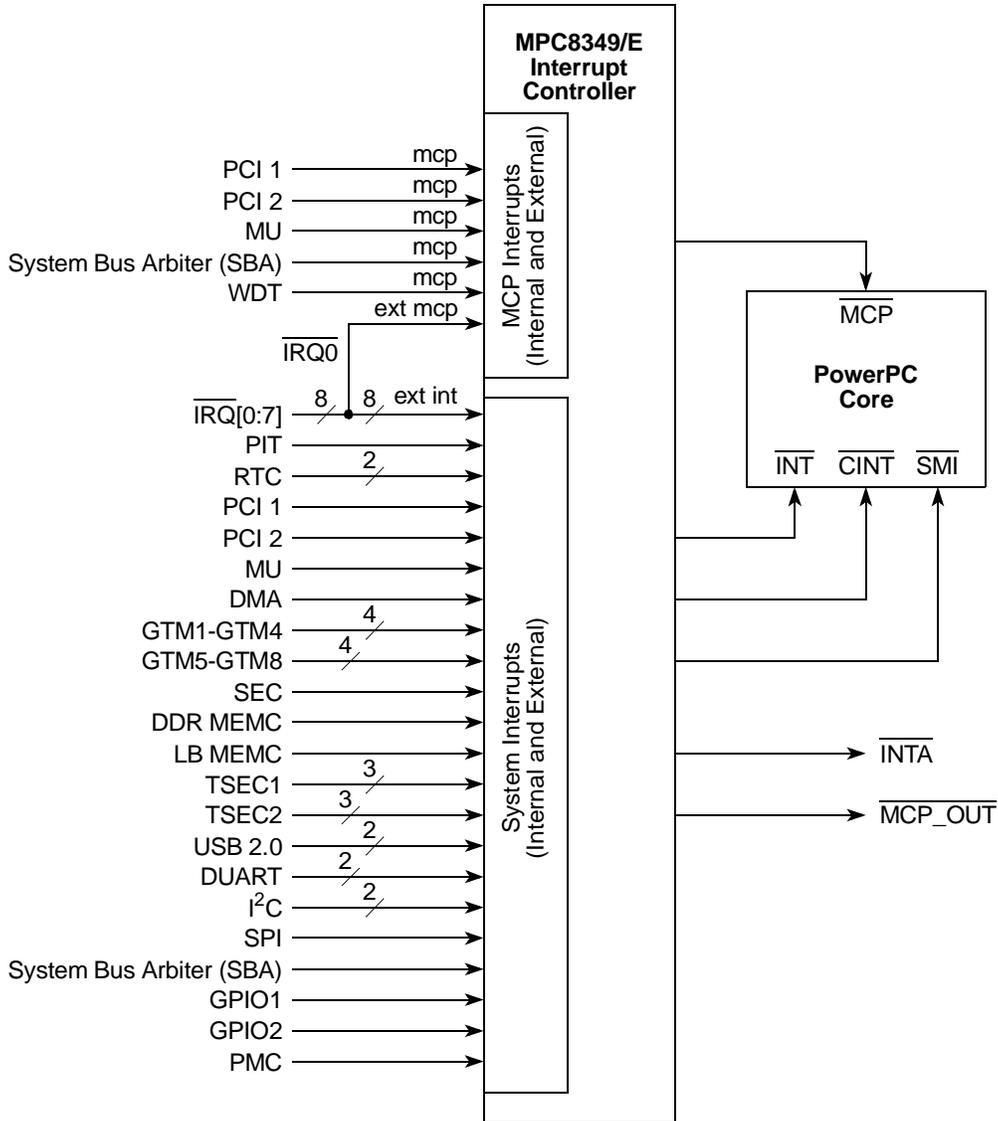


Figure 15. MPC83xx IPIC Structure

11 Clocks and Timers

The MPC83xx implements all of the clocking and timing functionality found on the MPC82xx family along with some enhancements. The primary MPC83xx device clocking and timing features are as follows:

- General-purpose timers
- Real-time clock
- Periodic interrupt timer

The following sections of the document highlight any differences in the programming model between the clocks and timers when moving from the MPC82xx to the MPC834x.

11.1 General-Purpose Timers

The MPC83xx general-purpose timer module is functionally similar to that of the MPC82xx. The MPC83xx supports two general-purpose timer modules, each providing four 16-bit timers for a total of eight 16-bit timers. The MPC82xx supports a maximum of four 16-bit timers.

The programming model for the MPC82xx and MPC83xx timer modules is also very similar. Table 4 list the corresponding registers for the MPC82xx and MPC83xx general-purpose timer modules. Note that the registers are identical apart from the following exceptions:

- The MPC82xx supports only two $\overline{\text{TGATE}}_x$ external pins. $\overline{\text{TGATE}}_{1-2}$ and $\overline{\text{TGATE}}_{3-4}$ supply gate/restart control for timers 1 and 2, 3 and 4, respectively. However, the MPC83xx supplies four individual $\overline{\text{TGATE}}$ signals for each 16-bit timer in the module. The MPC82xx TGCR1[GM1] and TGCR2[GM2] control the gate mode for each $\overline{\text{TGATE}}_x$ signal. The equivalent register bits for the MPC83xx are GTCFR1[GM1, GM2] and GTCFR2[GM3, GM4] The two extra fields (GM2 and GM4) are reserved in the MPC82xx programming model.
- The MPC826x general-purpose timers 1 and 2 can be cascaded to form a 32-bit counter, as can timers 3 and 4. On the MPC83xx, all four timers to be cascaded into one 64-bit counter by setting GTCFR2[SCAS] equal to 1. SCAS is a reserved field within the MPC82xx implementation.

Table 4. MPC82xx/MPC83xx General-Purpose Timer Register Comparison

Processor	Configuration Register	Mode Register	Reference Register	Capture Register	Counter Registers	Event Register
MPC82xx	TGCR1-2	TMR1-4	TRR1-4	TCR1-4	TCN1-4	TER1-4
MPC83xx	GTCFR1-2	GTMDR1-4	GTRFR1-4	GTCPR1-4	GTCNR1-4	GTEVR1-4

11.1.1 General-Purpose Timer Clocking

The internal clock input for the general-purpose timers also differs between the MPC82xx and MPC83xx families. The MPC82xx uses the internal clock CLKIN or CLKIN/16 to clock the timers, whereas the MPC83xx uses CSB_CLK or CSB_CLK/16. Also, the MPC83xx provides two prescale options to reduce the internal clock rate in comparison to the one in MPC82xx. The MPC83xx prescale values are programmed through the global timer prescale registers (GTPSR1-4) and the global timer mode register GTMDR[SPS]. For both the MPC82xx and MPC83xx, TIN is used as the external clock.

11.2 Real-Time Clock

The real-time clock (RTC) is used for the time and calendar generation. The equivalent function is provided by time counter (TMCNT) with the MPC82xx. Although the register definitions for the MPC83xx RTC and MPC82xx TMCNT differ, their functionality is almost identical. They have two functions:

- Generate an interrupt every second
- Generate an interrupt when the counter value reaches the value in the alarm register (TMCNTAL for MPC82xx and RTALR for MPC83xx)

Table 5 illustrates the equivalent RTC control register fields moving from the MPC82xx to the MPC83xx.

Table 5. MPC82xx/MPC83xx Real Time Clock Register Comparison

Processor	Enable	Second Mask	Alarm Mask	Second Event	Alarm Event
MPC82xx	TMCNTSC[TCE]	TMCNTSC[SIE]	TMCNTSC[ALE]	TMCNTSC[SEC]	TMCNTSC[ALR]
MPC83xx	RTCNR[CLEN]	RTCNR[SIM]	RTCNR[AIM]	RTCNR[SIF]	RTCNR[AIF]

Although the differences between the MPC82xx and the MPC83xx RTC are minimal, be aware of the following:

- The MPC82xx TMCNT input clock uses the internal signal `timersclk` that should be set to a fixed value of 8.192 KHz. The `timersclk` clock signal itself can be derived either from external pin or the internal BRG. The MPC83xx RTC input is the external pin `RTC_CLK` and should be connected to 32.768 KHz crystal. RTC can also be programmed to use the CSB bus clock as the clock source.
- The MPC82xx TMCNT count is reset to 0 at `/PORESET` but is not affected by `HRESET` or `SRESET`. However, the MPC83xx RTC count is reset to 0 at `HRESET` but is not affected by `SRESET`.
- The MPC82xx time count register (TMCNT) contains the current value of the time counter. This register is read/write and you can initialize it to a desired value before enabling the timer. This design has changed because the MPC83xx RTC real-time count register (RTCTR) allows only reads from this register. Write the desired initial value to the real-time count load register (RTLDR) and then enable the RTC. The RTLDR is loaded into RTCTR when the count starts.

11.3 Periodic Interrupt Timer

Both the MPC83xx and the MPC82xx have periodic interrupt timers (PIT) to provide regular timed interrupts for the real-time operating system and application software. [Table 6](#) lists the register fields of the two device families that have the same functionality

Table 6. MPC82xx/MPC83xx Periodic Interrupt Timer Register Comparison

Processor	Enable	Interrupt Mask	Interdit Event
MPC82xx	PISCR[PTE]	PISCR[PIE]	PISCR[PS]
MPC83xx	PTCNR[CLEN]	PTCNR[PIM]	PTCNR[SIF]

The main functional differences between the MPC82xx and the MPC83xx PIT implementations are as follows:

- The MPC82xx PIT is 16 bits wide. However, the MPC83xx has a 32-bit wide version.
- The MPC82xx PIT input clock uses `timersclk` that should be set to a fixed value of 8.192 KHz. The `timersclk` signal itself can be derived either from external pin or the internal BRG. The MPC83xx PIT shares the same external input pin as the RTC (`RTC_CLK`) discussed previously. `RTC_CLK` should be connected to 32.768 KHz crystal source to provide a 1sec interrupt period. The MPC83xx PIT can also be programmed to use the CSB bus clock as the clock source.

11.4 Clock Architecture

The MPC83xx bus architecture provides increased internal clocking flexibility in comparison to the MPC82xx. The MPC83xx provides high level internal clock flexibility via four specific DLLs to the following:

- Coherent system bus (CSB)
- DDR controller internal clock
- Local bus interface unit (LBIU)
- QUICC Engine (MPC835x/6x only)

NOTE

A second PLL inside the e300 core multiplies up the CSB clock frequency to create the internal clock for the e300 core.

This multiple clocking architecture for the MPC83xx greatly improves complete system management over the MPC82xx family. For example, for traffic-intensive designs in which the e300 core does not run at higher frequencies (power) but the MPC8360 QUICC Engine does, the user can program the clocking logic to provide this functionality. With the MPC82xx family this is not possible because the core/CPM/bus ratios are of a fixed range.

How to Reach Us:

Home Page:

www.freescale.com

email:

support@freescale.com

USA/Europe or Locations Not Listed:

Freescale Semiconductor
Technical Information Center, CH370
1300 N. Alma School Road
Chandler, Arizona 85224
(800) 521-6274
480-768-2130
support@freescale.com

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)
support@freescale.com

Japan:

Freescale Semiconductor Japan Ltd.
Headquarters
ARCO Tower 15F
1-8-1, Shimo-Meguro, Meguro-ku
Tokyo 153-0064, Japan
0120 191014
+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
(800) 441-2447
303-675-2140
Fax: 303-675-2150
LDCForFreescaleSemiconductor
@hibbertgroup.com

Information in this document is provided solely to enable system and software implementers to use Freescale Semiconductor products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits or integrated circuits based on the information in this document.

Freescale Semiconductor reserves the right to make changes without further notice to any products herein. Freescale Semiconductor makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale Semiconductor assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. "Typical" parameters which may be provided in Freescale Semiconductor data sheets and/or specifications can and do vary in different applications and actual performance may vary over time. All operating parameters, including "Typicals" must be validated for each customer application by customer's technical experts. Freescale Semiconductor does not convey any license under its patent rights nor the rights of others. Freescale Semiconductor products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the Freescale Semiconductor product could create a situation where personal injury or death may occur. Should Buyer purchase or use Freescale Semiconductor products for any such unintended or unauthorized application, Buyer shall indemnify and hold Freescale Semiconductor and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that Freescale Semiconductor was negligent regarding the design or manufacture of the part.

Freescale™, the Freescale log, and PowerQUICC are trademarks of Freescale Semiconductor, Inc. The product described contains a PowerPC processor core. The PowerPC name is a trademark of IBM Corp. and used under license. All other product or service names are the property of their respective owners.

© Freescale Semiconductor, Inc., 2006.

Document Number: AN2915

Rev. 1

3/2006