MOTOROLA
Semiconductor Products Sector

Order Number: MPC107TS/D
Rev. 0, 9/1999

PowerPC™

Advance Information

MPC107 PCI Bridge/Memory Controller Technical Summary

This document provides an overview of the MPC107 PCI bridge/memory controller (PCIB/MC) for high-performance embedded systems. The MPC107 is a cost-effective, general-purpose PCI bridge/memory controller for applications using PCI in networking infrastructure, telecommunications, and other embedded markets. It can be used in applications such as network routers and switches, mass storage subsystems, network appliances, and print and imaging systems.

For errata or revisions to this document, refer to the website at http://www.motorola.com/powerpc.

1.1 MPC107 PCI Bridge/Memory Controller Features

The MPC107 provides an integrated high-bandwidth, high-performance interface between up to two 60x processors, the PCI bus, and main memory. This section summarizes the features of the MPC107. Major features of the MPC107 are as follows:

- Memory interface
  - — 64-/32-bit 100-MHz bus
  - — Programmable timing supporting either FPM DRAM, EDO DRAM or SDRAM
  - — High-bandwidth bus (32-/64-bit data bus) to DRAM

  - — Supports one to eight banks of 4-, 16-, 64-, or 128-Mbit memory devices, and up to four banks of 256-Mbit SDRAM devices
  - — Supports 1-Mbyte to 1-Gbyte DRAM memory
  - — 144 Mbytes of ROM space
  - — 8-, 32-, or 64-bit ROM
  - — Write buffering for PCI and processor accesses
  - — Supports normal parity, read-modify-write (RMW), or ECC
  - — Data-path buffering between memory interface and processor
  - — Low-voltage TTL logic (LVTTL) interfaces
  - — Port X: 8-, 32-, or 64-bit general-purpose I/O port using ROM controller interface with programmable address strobe timing
- 32-bit PCI interface operating up to 66 MHz
  - — PCI 2.1-compliant
  - — PCI 5.0-V tolerance
  - — Support for PCI locked accesses to memory
  - — Support for accesses to PCI memory, I/O, and configuration spaces
  - — Selectable big- or little-endian operation
  - — Store gathering of processor-to-PCI write and PCI-to-memory write accesses
  - — Memory prefetching of PCI read accesses
  - — Selectable hardware-enforced coherency
  - — PCI bus arbitration unit (five request/grant pairs)
  - — PCI agent mode capability
  - — Address translation unit
  - — Some internal configuration registers accessible from PCI
- Two-channel integrated DMA controller (writes to ROM/Port X not supported)
  - — Supports direct mode or chaining mode (automatic linking of DMA transfers)
  - — Supports scatter gathering—read or write discontinuous memory
  - — Interrupt on completed segment, chain, and error
  - — Local-to-local memory
  - — PCI-to-PCI memory
  - — PCI-to-local memory
  - — PCI memory-to-local memory

- Message unit
  - — Two doorbell registers
  - — An extended doorbell register mechanism that facilitates interprocessor communication through interrupts in a dual-local-processor system
  - — Two inbound and two outbound messaging registers
  - — I_2O message controller
- I^2C controller with full master/slave support (except broadcast all)
- Embedded programmable interrupt controller (EPIC)
  - — Five hardware interrupts (IRQs) or 16 serial interrupts
  - — Four programmable timers
- Integrated PCI bus, CPU, and SDRAM clock generation
- Programmable PCI bus, 60x, and memory interface output drivers
- Dynamic power management—Supports 60x nap, doze, sleep, and suspend modes
- Programmable input and output signals with watchpoint capability
- Built-in PCI bus performance monitor facility
- Debug features
  - — Error injection/capture on data path
  - — IEEE 1149.1 (JTAG)/test interface
- Processor interface
  - — Supports up to two PowerPC™ microprocessors with 60x bus interface
  - — Supports various operating frequencies and bus divider ratios
  - — 32-bit address bus, 64/32-bit data bus supported at 100 MHz
  - — Supports full memory coherency
  - — Supports optional local bus slave
  - — Decoupled address and data buses for pipelining of 60x accesses
  - — Store gathering on 60x-to-PCI writes
  - — Concurrent transactions on 60x and PCI buses supported

1.2 MPC107 PCI/MC Applications

The MPC107 can be used in either a system host configuration or as a peripheral device. For system applications where cost, space, and power consumption are critical parameters, yet performance cannot be sacrificed, the MPC107 provides a complete solution. The MPC107 is shown in Figure 1-1 as a host bridge.

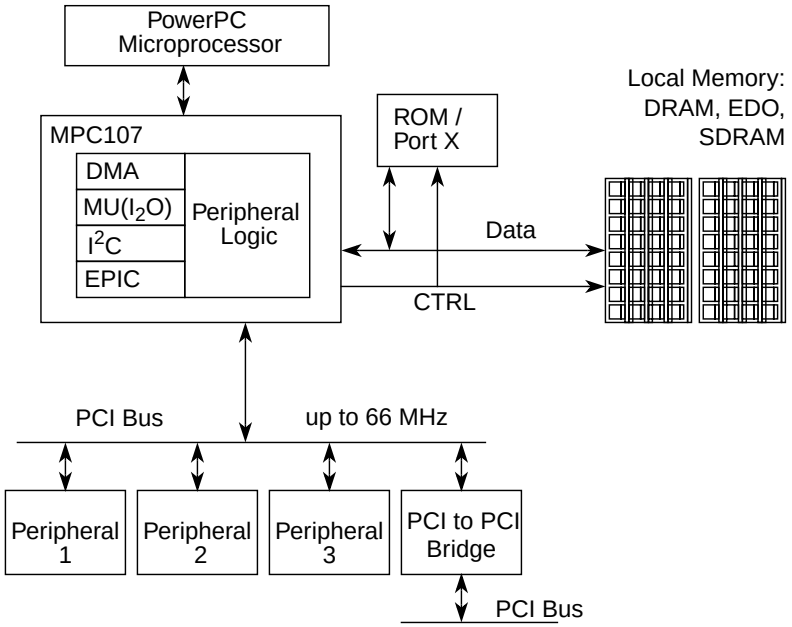

Figure 1-1. System Using MPC107 as a Host Bridge

With the embedded enhancements provided in the MPC107, it is possible to use it in distributed I/O processor applications as shown in Figure 1-2.

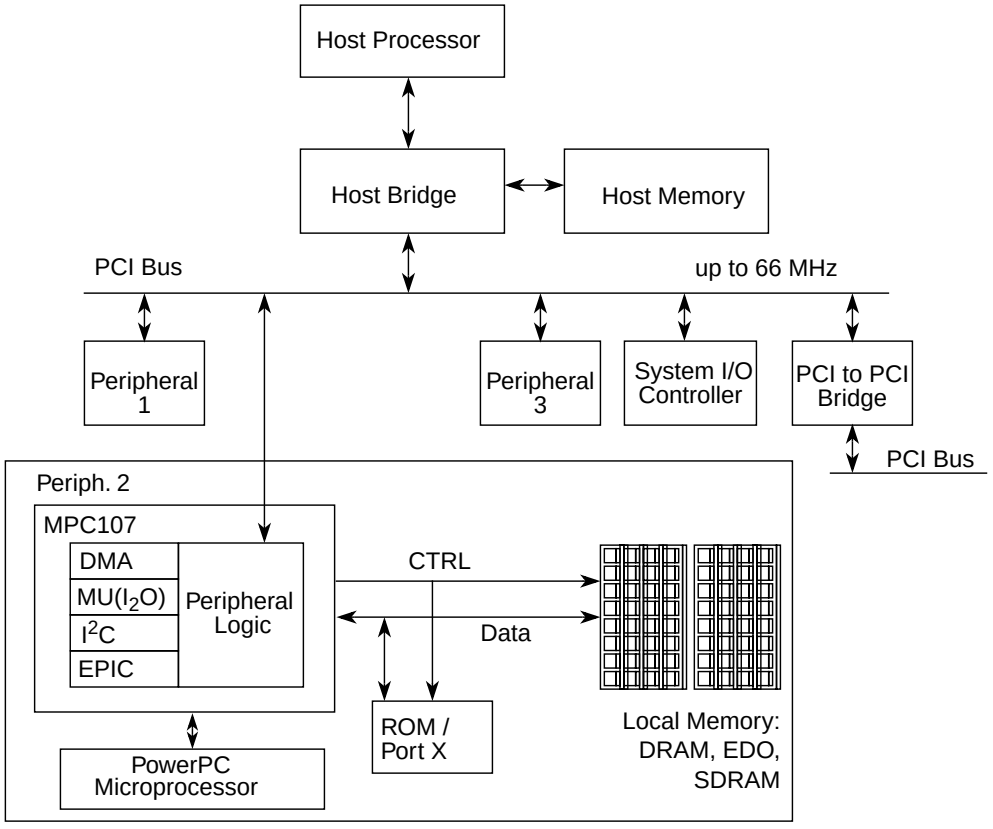

Figure 1-2. Embedded System Using the MPC107 as a Bridge to a Distributed I/O Processor

The MPC107 can also be used with a peripheral processing device as shown in Figure 1-3. In this case, the PCI-to-PCI bridge shown Figure 1-3 could be of the PCI type 0 variety. The MPC107 would not be part of the system configuration map. This configuration is useful in applications such as RAID controllers or multi-port network controllers where the I/O devices shown are SCSI controllers or Ethernet controllers, respectively.

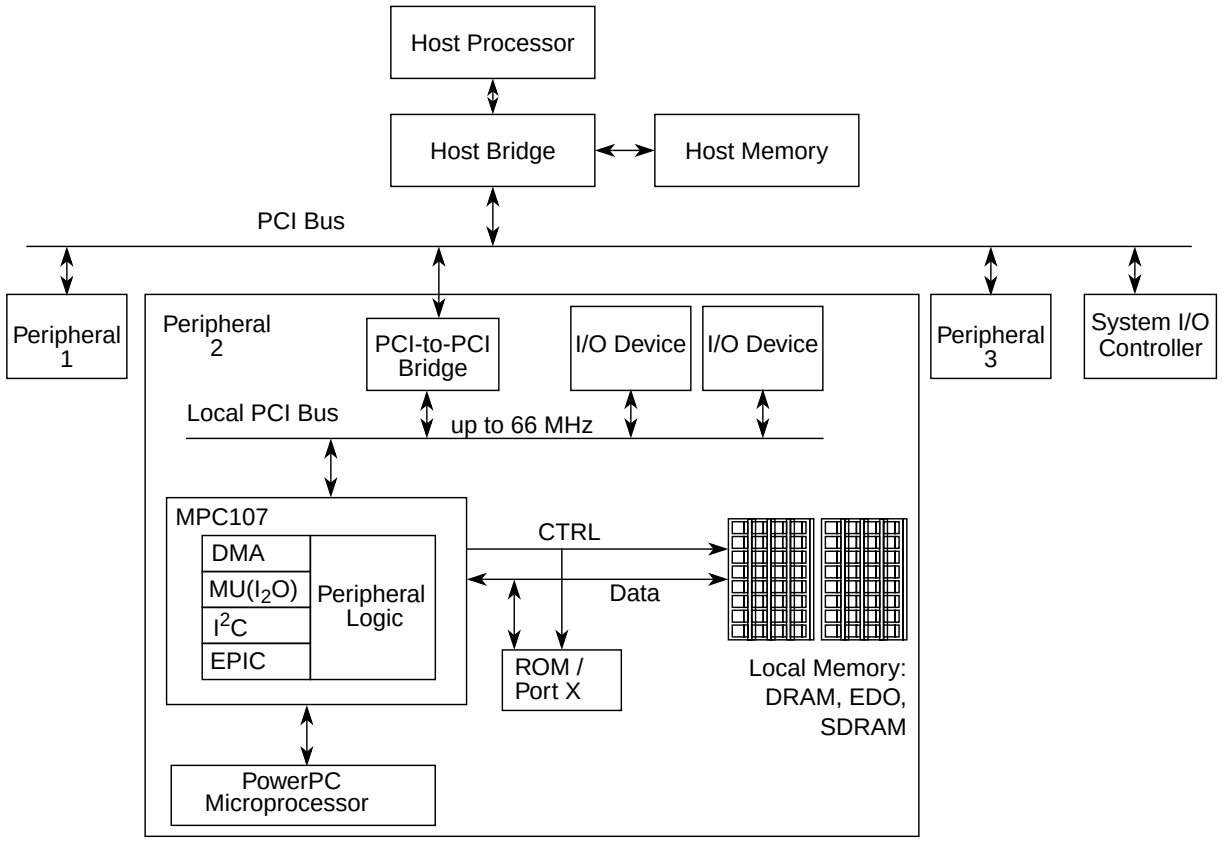

Figure 1-3. Embedded System Using MPC107 with a Peripheral Processor

The processor bus interface (60x) of the MPC107 contains all of the necessary arbitration and control logic to communicate with up to two PowerPC microprocessors in a symmetric multiprocessing environment. In addition, the MPC107 also has a side-band mode to allow an alternate bus slave to capture address tenures. This application is show in Figure 1-4.

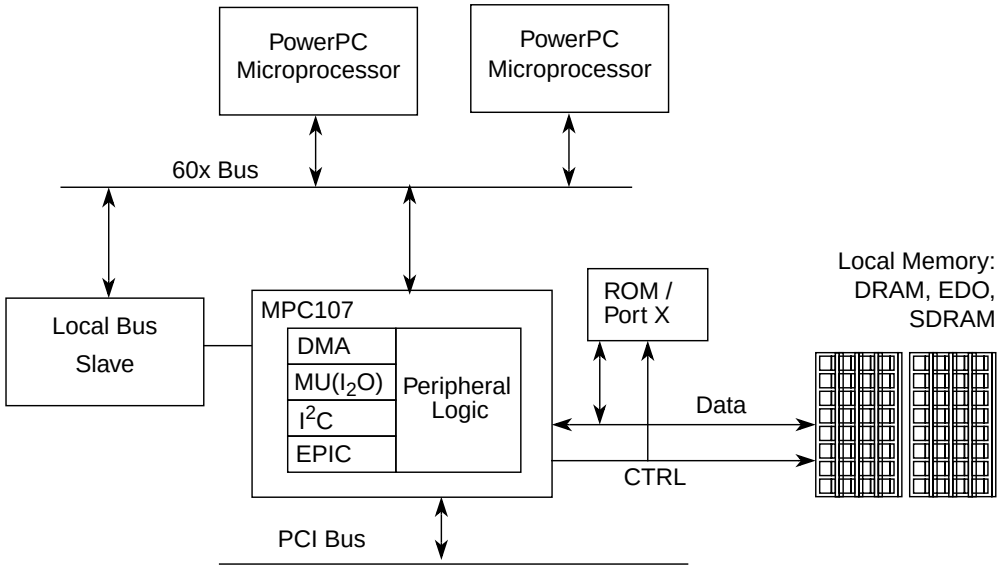

Figure 1-4. Multiprocessor System Using the MPC107 and a Local Bus Slave

1.3 MPC107 Major Functional Blocks

The MPC107 integrates a PCI bridge, memory controller, DMA controller, EPIC interrupt controller/timers, a message unit with an Intelligent Input/Output (I_2O) message controller, and an Inter-Integrated Circuit (I^2C) controller. The integration reduces the overall packaging requirements and the number of discrete devices required for an embedded system.

Figure 1-5 shows the major functional units within the MPC107. Note that this is a conceptual block diagram intended to show the basic features rather than an attempt to show how these features are physically implemented.

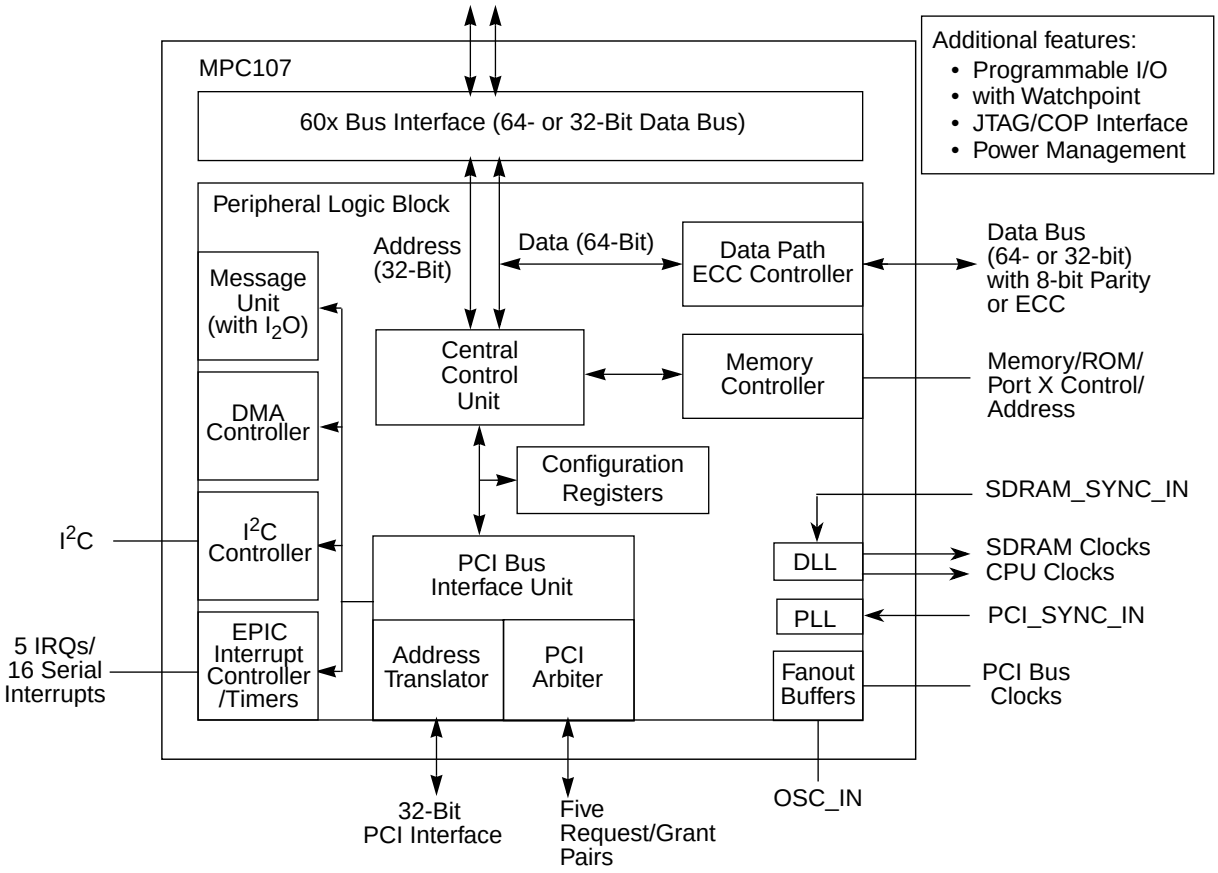

Figure 1-5. MPC107 Block Diagram

1.3.1 60x Processor Interface

The MPC107 supports a programmable interface to a variety of PowerPC microprocessors operating at select bus speeds. The 60x address bus is 32 bits wide and the data bus is configurable to be 64 or 32 bits wide. The 60x processor interface of the MPC107 uses a subset of the 60x bus protocol, supporting single-beat and burst data transfers. The address and data buses are decoupled to support pipelined transactions.

In this document, the term ‘60x’ is used to denote a 32-bit microprocessor from the PowerPC architecture family that conforms to the bus interface of the MPC603e, MPC740, MPC750, or MPC7400 microprocessors. 60x processors implement the PowerPC architecture as it is specified for 32-bit addressing, which provides 32-bit effective (logical) addresses, integer data types of 8, 16, and 32 bits, and floating-point data types of 32 and 64 bits (single-precision and double-precision).

Two signals on the MPC107, local bus slave claim ($\overline{\text{LBCLAIM}}$) and data bus grant local bus slave ($\overline{\text{DBGLB}}$), are provided for an optional local bus slave. However, the local bus slave must be capable of generating the transfer acknowledge ($\overline{\text{TA}}$) signal to interact with the 60x processor(s).

Depending on the system implementation, the processor bus may operate at the PCI bus clock rate, or at a multiple of the PCI bus clock rate (determined at reset). The 60x processor bus is synchronous, with all timing relative to the rising edge of the 60x bus clock.

When two 60x processors are used, the two sets of bus request, bus grant, and data bus grant signals allow for arbitration between the 60x processors. The 60x processors share all 60x interface signals of the MPC107, except the bus arbitration signals.

1.3.2 Memory System Interface

The MPC107 memory interface controls processor and PCI interactions to main memory. It supports a variety of DRAM, and flash or ROM configurations as main memory. The MPC107 supports fast page mode (FPM), extended data out (EDO) and synchronous DRAM (SDRAM). The maximum supported memory size is 1 Gbyte of DRAM or SDRAM and 144 Mbytes of ROM/flash. SDRAM must comply with the JEDEC SDRAM specification.

The MPC107 implements Port X, a memory bus interface that facilitates the connection of general-purpose I/O devices. The Port X functionality allows the designer to connect external registers, communication devices, and other such devices directly to the MPC107. Some devices may require a small amount of external logic to properly generate address strobes, chip selects, and other signals.

The MPC107 is designed to control a 32-bit or 64-bit data path to main memory DRAM or SDRAM. For a 32-bit data path, the MPC107 can be configured to check and generate byte parity using four parity bits. For a 64-bit data path, the MPC107 can be configured to support parity or ECC checking and generation with eight parity/syndrome bits checked and generated. Note that the data bus width (32- or 64-bit) chosen at reset for the 60x bus interface is also used for the memory interface.

The MPC107 supports DRAM bank sizes from 1 to 128 Mbytes, SDRAM bank sizes of 1 to 256 Mbytes, and provides bank start address and end address configuration registers. Note that the MPC107 does not support mixed DRAM/SDRAM configurations. The MPC107 can be configured so that appropriate row and column address multiplexing occurs according to the accessed memory bank. Addresses are provided to DRAM and SDRAM through a 13-bit interface for DRAM and a 15-bit interface for SDRAM.

The memory bus width is configured at reset as 32- or 64-bits wide, depending on the setting of the MDL0 reset configuration signal. Four chip selects, one write enable, one output enable, and up to 21 address signals are provided for ROM/flash systems.

1.3.3 Peripheral Component Interconnect (PCI) Interface

The PCI interface for the MPC107 is compliant with the *Peripheral Component Interconnect Specification* Revision 2.1. The PCI interface provides mode-selectable, big- to little-endian conversion. The MPC107 provides an interface to the PCI bus running at speeds up to 66 MHz.

The MPC107's PCI interface can be configured as host or agent. In host mode, the interface acts as the main memory controller for the system and responds to all host memory transactions.

In agent mode, the MPC107 can be configured to respond to a programmed window of PCI memory space. A variety of initialization modes are provided to boot the device.

1.3.3.1 PCI Bus Arbitration Unit

The MPC107 contains a PCI bus arbitration unit, which eliminates the need for an external unit, thus lowering system complexity and cost. It has the following features:

- Five external arbitration signal pairs. The MPC107 is the sixth member of the arbitration pool.
- The bus arbitration unit allows fairness as well as a priority mechanism.
- A two-level round-robin scheme is used in which each device can be programmed within a pool of a high- or low-priority arbitration. One member of the low-priority pool is promoted to the high-priority pool. As soon as it is granted the bus, it returns to the low-priority pool.
- The unit can be disabled to allow a remote arbitration unit to be used.

1.3.3.2 Address Maps and Translation

The MPC107's processor bus interface supports memory-mapped accesses. The address space is divided between memory and PCI according to one of two allowable address maps—map A and map B. Note that the support of map A is provided for backward compatibility only. It is strongly recommended that new designs use map B because map A may not be supported in future devices.

An inbound and outbound PCI address translation mechanism is provided to support the use of the MPC107 in agent mode. Note that address translation is supported only for agent mode; it is not supported when the MPC107 is operating in host mode. Also note that since agent mode is supported only for address map B, address translation is supported only for address map B.

When the MPC107 is configured to be a PCI agent, the amount of local memory visible to the system is programmable. In addition, it may be necessary to map the local memory to a different system memory address space. The address translation unit handles the mapping of both inbound and outbound transactions for these cases.

1.3.3.3 Byte Ordering

The MPC107 allows the processor to run in either big- or little-endian mode (except for the initial boot code which must run in big-endian mode).

1.3.3.4 PCI Agent Capability

In certain applications, the embedded system architecture dictates that the MPC107 bridges to a peripheral processor. In this case, the peripheral logic must not act like a host bridge for the PCI bus. Instead it functions as a configurable device that is accessed by a host bridge. This capability allows multiple MPC107 devices to coexist with other PCI peripheral devices on a single PCI bus. The MPC107 has PCI 2.1- compliant configuration capabilities.

1.3.4 DMA Controller

The integrated DMA controller contains two independent units. Note that the DMA writing capability for local memory is available for DRAM and SDRAM, but writing is not available for the ROM/Port X interface. Each DMA unit is capable of performing the following types of transfers:

- PCI-to-local memory
- Local-to-PCI memory
- PCI-to-PCI memory
- Local-to-local memory

The DMA controller allows chaining through local memory-mapped chain descriptors. Transfers can be scatter gathered and misaligned. Interrupts are provided on completed segment, chain, and error conditions.

1.3.5 Message Unit (MU)

Many embedded applications require handshake algorithms to pass control, status, and data information from one owner to another. This is made easier with doorbell and message registers. The MPC107 has a message unit (MU) that implements doorbell and message registers as well as an I_2O interface. The MU has many conditions that can cause interrupts and uses the EPIC unit to signal interrupts to the PCI interface and interrupts to the processor.

1.3.5.1 Doorbell Registers

The MPC107 MU contains one 32-bit inbound doorbell register and one 32-bit outbound doorbell register. The inbound doorbell register allows a remote processor to set a bit in the register from the PCI bus. This, in turn, generates an interrupt to the local processor.

The local processor can write to the outbound register, causing the outbound interrupt signal $\overline{\text{INTA}}$ to assert, thus interrupting the host processor. Once $\overline{\text{INTA}}$ is generated, it can be cleared only by the host processor by writing ones to the bits that are set in the outbound doorbell register.

1.3.5.2 Extended Doorbell Register Facility

The MPC107 MU also contains an extended doorbell register mechanism that facilitates interprocessor communication through interrupts in a dual-local-processor system. For example, processor 0 can write to the EDBW1S register to generate an interrupt to processor 1. Processor 1 can then clear the interrupt by writing to the EDBW1C register. The same process can be used if processor 1 needs to interrupt processor 0. Use of this facility requires that the rest of the MU be disabled.

1.3.5.3 Inbound and Outbound Message Registers

The MPC107 contains two 32-bit inbound message registers and two 32-bit outbound message registers. The inbound registers allow a remote host or PCI master to write a 32-bit value, causing an interrupt to the local processor. The outbound registers allow the local processor to write an outbound message which causes the outbound interrupt signal $\overline{\text{INTA}}$ to assert.

1.3.5.4 Intelligent Input/Output Controller (I_2O)

The intelligent I/O specification is an open standard that defines an abstraction layer interface between the OS and subsystem drivers. Messages are passed between the message abstraction layer from one device to another.

The I_2O specification describes a system as being made up of host processors and input/output platforms (IOPs). The host processor is a single processor or a collection of processors working together to execute a homogenous operating system. An IOP consists of a processor, memory, and I/O interfaces. The IOP functions separately from other processors within the system to handle system I/O functions.

The I_2O controller of the MU enhances communication between hosts and IOPs within a system. There are two paths for messages—an inbound queue is used to transfer messages from a remote host or IOP to the local processor, and an outbound queue is used to transfer messages from the local processor to the remote host. Each queue is implemented as a pair of FIFOs. The inbound and outbound message queues each consists of a free_list FIFO and a post_list FIFO.

Messages are transferred between the host and the IOP using PCI memory-mapped registers. The MPC107's I_2O controller facilitates moving the messages to and from the inbound and outbound registers and local IOP memory. Interrupts signal the host and IOP to indicate the arrival of new messages.

1.3.6 Inter-Integrated Circuit (I^2C) Controller

The I^2C serial interface has become an industry de facto standard for communicating with low-speed peripherals. Typically, it is used for system management functions and EEPROM support. The MPC107 contains an I^2C controller with full master and slave functionality.

1.3.7 Embedded Programmable Interrupt Controller (EPIC)

The integrated embedded programmable interrupt controller (EPIC) of the MPC107 reduces the overall component count in embedded applications. The EPIC unit is designed to collect external and internal hardware interrupts, prioritize them, and deliver them to the local processor.

The module operates in one of three modes:

- In direct mode, five level- or edge-triggered interrupts can be connected directly to an MPC107.
- In pass-through mode, interrupts detected at the IRQ0 input are passed directly (with logic inversion) to the $\overline{\text{INT}}$ output signal. Also in this case, interrupts generated by the I_2O, I^2C, and DMA controllers are passed to the $\overline{\text{L\_INT}}$ output signal.
- The MPC107 provides a serial delivery mechanism for when more than five external interrupt sources are needed. The serial mechanism allows for up to 16 interrupts to be serially scanned into the MPC107. This mechanism increases the number of interrupts without increasing the number of pins.

The outbound interrupt request signal, $\overline{\text{L\_INT}}$, is used to signal interrupts to the host processor when the MPC107 is configured for agent mode. The MPC107 EPIC includes four programmable timers that can be used for system timing or for generating periodic interrupts.

1.3.8 Integrated PCI Bus, CPU, and SDRAM Clock Generation

There are two PCI bus clocking solutions directed towards different system requirements. For systems where the MPC107 is the host controller with a minimum number of clock loads, five clock fanout buffers are provided on-chip.

For systems requiring more clock fan out or where the MPC107 is an agent device, external clock buffers may be used.

The MPC107 provides an on-chip delay-locked loop (DLL) that supplies the external memory bus clock signals to SDRAM banks and also supplies three CPU clock outputs that are synchronized to the SDRAM clocks. The memory bus clock signals are of the same frequency and synchronous with the processor clock signals.

The four SDRAM clock outputs are generated by the internal DLL and can account for the trace length between SDRAM_SYNC_OUT signal and the SDRAM_SYNC_IN signal.

The MPC107 requires a single clock input signal, PCI_SYNC_IN, which can be driven by the PCI clock fan-out buffers—specifically the PCI_SYNC_OUT output. PCI_SYNC_IN can also be driven by an external clock driver.

PCI_SYNC_IN is driven by the PCI bus frequency. An internal PLL, using PCI_SYNC_IN as a reference, generates an internal *sys-logic-clk* signal that is used for the internal logic. The processor (60x) bus clock frequency on the CPU_CLK[0:2] outputs is configured at reset (by the MPC107 PLL configuration signals (PLL_CFG[0:3])) to be a multiple of the PCI_SYNC_IN frequency.

1.4 Power Management

The MPC107 provides program-controllable power reduction modes for progressive reduction of power consumption and a system hardware mechanism for further power reduction.

The MPC107 provides hardware support for three levels of programmable power reduction—doze, nap, and sleep modes; each is invoked by programming configuration registers.

The MPC107 also has a fourth power reduction mode invoked by the assertion of the external $\overline{\text{SUSPEND}}$ signal. In suspend mode, the MPC107 may have its clock input and PLL shut down for additional power savings. Memory refresh can be accomplished in two ways—either by using self-refresh mode DRAMs or by using the RTC input. To exit the suspend mode, the system clock must be turned on in sufficient time to restart the PLL. After this time, $\overline{\text{SUSPEND}}$ may be negated. In suspend mode, all bidirectional and output signals (except the memory refresh-related signals if RTC refresh is being used) are at high impedance and all input signals except the PLL_CFG[0:3] signals, $\overline{\text{HRST}}$, and $\overline{\text{SUSPEND}}$, are ignored. Note that the $\overline{\text{SUSPEND}}$ signal is monitored for negation during suspend mode.

The MPC107 is fully static, allowing internal logic states to be preserved during all power-saving modes.

Table 1-1 summarizes the programmable power-saving modes for the MPC107.

Table 1-1. MPC107 Power Modes Summary

PM Mode	Functioning Units	Activation Method	Full-Power Wake Up Method
Full power	All units active	—	—
Doze	PCI address decoding and bus arbiter System RAM refreshing 60x bus request and NMI monitoring EPIC unit I^2C unit PLL	Controlled by software (write to PMCR1)	PCI access to memory 60x bus request Assertion of NMI[1] Hard Reset

Table 1-1. MPC107 Power Modes Summary (Continued)

PM Mode	Functioning Units	Activation Method	Full-Power Wake Up Method
Nap	PCI address decoding and bus arbiter System RAM refreshing 60x bus request and NMI monitoring EPIC unit I^2C unit PLL	Controlled by software (write to PMCR1) and processor in nap or sleep mode ($\overline{QREQ}$ asserted)	PCI access to memory[2] 60x bus request[3] Assertion of NMI[1] Hard Reset
Sleep	PCI bus arbiter System RAM refreshing (can be disabled) 60x bus request and NMI monitoring EPIC unit I^2C unit PLL (can be disabled)	Controlled by software (write to PMCR1) and processor in nap or sleep mode ($\overline{QREQ}$ asserted)	60x bus request[3] Assertion of NMI[1] Hard Reset
Suspend	System RAM refreshing (can be disabled) PLL (can be disabled) $\overline{SUSPEND}$, PLL_CFG[0:3], and $\overline{HRST}$ signal monitoring	Assertion of $\overline{SUSPEND}$	$\overline{SUSPEND}$ negation Hard Reset

1 Programmable option based on value of PICR1[MCP_EN] = 1.

2 A PCI access to memory in nap mode causes $\overline{QACK}$ to negate while the MPC107 services the access. Additionally, some 60x processors (MPC740, MPC750, and MPC7400) will wake up and respond to the snoop transaction. After servicing the PCI access, the MPC107 automatically returns to the nap mode.

3 Programmable option for recognition of $\overline{BR1}$ (bus request from second local processor in a dual processor system) based on PMCR1[BR1_WAKE]

1.5 Programmable I/O Signals with Watchpoint

The MPC107 programmable I/O facility allows the system designer to monitor the 60x bus. Up to two watchpoints and their respective 4-bit countdown values can be programmed. When the programmed threshold of the selected watchpoint is reached, an external trigger signal is generated and the states of the 60x address, control, and data buses are latched into user-readable registers.

1.6 Debug Features

The MPC107 includes the following debug features:

- Error injection/capture on data path
- IEEE 1149.1 (JTAG)/test interface

1.6.1 Error Injection/Capture on Data Path

The MPC107 provides hardware to exercise and debug the ECC and parity logic by allowing the user to inject multi-bit stuck-at faults onto the peripheral logic or memory data/parity buses and to capture the data/parity output on receipt of an ECC or parity error.

1.6.2 IEEE 1149.1 (JTAG)/Test Interface

To facilitate system testing, the MPC107 provides a JTAG test access port that complies with the IEEE 1149.1 boundary-scan specification.

Motorola Literature Distribution Centers:

USA/EUROPE: Motorola Literature Distribution; P.O. Box 5405; Denver, Colorado 80217; Tel.: 1-800-441-2447 or 1-303-675-2140;
World Wide Web Address: http://ldc.nmd.com/

JAPAN: Nippon Motorola Ltd SPD, Strategic Planning Office 4-32-1, Nishi-Gotanda Shinagawa-ku, Tokyo 141, Japan Tel.: 81-3-5487-8488

ASIA/PACIFIC: Motorola Semiconductors H.K. Ltd Silicon Harbour Centre 2, Dai King Street Tai Po Industrial Estate Tai Po, New Territories, Hong Kong

Mfax™: RMFAX0@email.sps.mot.com; TOUCHTONE 1-602-244-6609; US & Canada ONLY (800) 774-1848;
World Wide Web Address: http://sps.motorola.com/mfax

INTERNET: http://motorola.com/sps

Technical Information: Motorola Inc. SPS Customer Support Center 1-800-521-6274; electronic mail address: crc@wmkmail.sps.mot.com.

Document Comments: FAX (512) 895-2638, Attn: RISC Applications Engineering.

World Wide Web Addresses: http://www.motorola.com/PowerPC
http://www.motorola.com/netcomm
http://www.motorola.com/Coldfire

MOTOROLA

MPC107TS/D