Application Note

AN2540/D
Rev. 0.1, 6/2003

Synchronizing Instructions for PowerPC™ Instruction Set Architecture

MOTOROLA intelligence everywhere™ | digital dna™

The architecture for the PowerPC™ Instruction Set provides two types of synchronizing instructions: *context-synchronizing* and *execution-synchronizing* instructions. All of the information in this application note is covered in more detail in *PowerPC™ Microprocessor Family: The Programming Environments for 32-Bit Processors*.

1 Context-Synchronizing Instructions

Context-synchronizing instructions for PowerPC include **isync**, **sc**, and **rfi**. In addition, exceptions can cause context synchronization. These instructions can be used to ensure that the effects of all previously issued instructions are in place before a context switch, and that the context switch takes effect for instructions after the switch. Context synchronization should be performed when changing the values in certain fields of certain processor registers, as shown in Table 1.

The context-synchronizing instructions ensure that the following conditions occur:

- Instruction dispatching is halted. The **sc** instruction also ensures that no higher priority exception exists.
- All previously issued instructions have completed, at least to a point where they are no longer able to cause an exception. However, memory accesses that these instructions cause need not have completed with respect to other processors and mechanisms. The **sync** instruction can be used to ensure that memory accesses are complete.
- Instructions that were previously issued complete in the context in which they were issued (privilege, protection, address translation).
- Instructions that are issued after the synchronizing instruction execute in the new context.

To ensure that context changes occur for instructions after the synchronization, the instruction queue is flushed and all these instructions are refetched with the new context in place.

NOTE

All context-synchronizing instructions are execution-synchronizing.

2 Execution-Synchronizing Instructions

Context-synchronizing instructions are most useful for ensuring that context switches take effect at the right time. Execution-synchronizing instructions are useful for providing a sequencing function in a program and for memory access synchronization on multiple processor implementations. These instructions can be used to ensure that previous instructions have apparently completed before subsequent instructions are executed without forcing a flush of prefetched instructions in the instruction queue. This feature is often useful for specifically ordering programs and for debugging. Programmers should be aware, however, that the instruction uses a significant amount of time to complete, and should not be used indiscriminately. For many applications that specifically order loads and stores, the **eieio** instruction may provide the desired effect at lower cost to performance.

Execution-only synchronizing instructions (those that are not also context-synchronizing) are different from context-synchronizing instructions in that they do not ensure that subsequent instructions execute in the context that the synchronizing instruction establishes. (When the instruction queue is not flushed, instructions after the synchronization in the queue execute in the old context). The new context takes effect sometime after the execution-synchronizing instruction completes. Like context-synchronizing instructions, all execution-synchronizing instructions halt dispatching and ensure that previous instructions have completed to the point where they cannot cause an exception. Execution-synchronizing instructions include **sync**, **mtmsr**, and all context-synchronizing instructions.

The **sync** instruction, besides synchronizing execution, can broadcast addresses on the bus and is sometimes used for synchronizing coherent memory with other processors. Unlike **isync**, **sync** forces all external accesses to complete with respect to other processors and mechanisms that access memory.

2.1 Synchronization Requirements for Instructions that Alter Context

Table 1 summarizes general information about synchronization requirements for different instructions that alter context. Individual processors occasionally have additional synchronization requirements or restrictions. See *PowerPC™ Microprocessor Family: The Programming Environments for 32-Bit Processors,* section 2.3.17, "Synchronization Requirements for Special Registers and for Lookaside Buffers" and your specific processor manual for complete information.

Table 1. Synchronization Requirements for Instructions that Alter Context

Instruction	Synchronization Required Before	Synchronization Required After
mtmsr[PR]	none	context
mtmsr[FP](instr access)	none	context
mtmsr[FE0,FE1](instr access)	none	context
mtmsr[SE,BE](instr access)	none	context
mtmsr[ME]	none	context
mtmsr[LE]	implementation dependent	implementation dependent
mtmsr[DR](data access)	none	context
mtmsr[IR](instr access)	none	context
mtmsr[POW](instr access)	implementation dependent	implementation dependent

Table 1. Synchronization Requirements for Instructions that Alter Context (continued)

Instruction	Synchronization Required Before	Synchronization Required After
lmtmsr[DABR](data access)	implementation dependent	implementation dependent
mtsr(data access)	context	context
mtsr(instr access)	none	context
mtsrin(data access)	context	context
mtsrin(instr access)	none	context
mtspr[SDR1]	**sync**	context
mtspr[DBAT](data access)	context	context
mtspr[IBAT](instr access)	none	context
mtspr[EAR](data access)	context	context
tlbie(data access)	context	context or **sync**
tlbie(instr access)	none	context or **sync**
tlbia(data access)	context	context or **sync**
tlbia(instr access)	none	context or **sync**

2.2 Hardware Implementation Registers (HIDs)

In addition to the instructions listed in Table 1, many processors have bits in the Hardware Implementation Registers (HIDs) that require some form of synchronization when they are changed. For example, many PowerPCs require synchronization before changing the bits that control cache enabling and locking for the instruction and data caches. Changing the DCE and DLOCK bits in HID0 for many processors requires a **sync** before a change, and setting the ICE and ILOCK bits in HID0 necessitates an **isync** before the change. Please read your processor's manual carefully for more information about synchronization requirements when changing bits in Hardware Implementation Registers.

NOTE

Some register fields require synchronization differently for data or instruction accesses.

3 Sources of Information

More information on this topic can be found in Motorola's PowerPC library and in the following publications:

- *PowerPC™ Microprocessor Family: The Programming Environments for 32-Bit Processors*
  - — Section 4.1.5 Synchronizing Instructions
  - — Section 6.1.2 Synchronization
  - — Section 2.3.17 Synchronization Requirements for Special Registers and for Lookaside Buffers
  - — Section 8.2 PowerPC Instruction Set
  - — Appendix E Synchronization Programming Examples (see listings for **sync**, **isync**, **rfi**, **sc**, **mtmsr**, and **eieio**)

- *MPC603e & EC603e User's Manual*
  - — Section 2.3.2.4 Synchronization
  - — Section 2.3.5.2 Memory Synchronization
  - — Section 6.3.3.2 Instruction Serialization
- *MPC750 User's Manual*
  - — Section 2.3.2.4 Synchronization
  - — Section 2.3.4.7 Memory Synchronization Instruction
  - — Section 4.4 Process Switching
  - — Section 6.3.2.2 Instruction Serialization

The manuals for the individual processors that are not listed contain details for a specific processor also. This information would be too cumbersome to list in this document. See your processor manual for complete information.

4 Revision History

Table 2 provides a revision history for this application note.

Table 2. Document Revision History

Rev. No.	Substantive Change(s)
0	Initial release
0.1	Nontechnical reformatting

THIS PAGE INTENTIONALLY LEFT BLANK

THIS PAGE INTENTIONALLY LEFT BLANK

THIS PAGE INTENTIONALLY LEFT BLANK

HOW TO REACH US:

USA/EUROPE/LOCATIONS NOT LISTED:

Motorola Literature Distribution
P.O. Box 5405, Denver, Colorado 80217
1-480-768-2130
(800) 521-6274

JAPAN:

Motorola Japan Ltd.
SPS, Technical Information Center
3-20-1, Minami-Azabu Minato-ku
Tokyo 106-8573 Japan
81-3-3440-3569

ASIA/PACIFIC:

Motorola Semiconductors H.K. Ltd.
Silicon Harbour Centre, 2 Dai King Street
Tai Po Industrial Estate, Tai Po, N.T., Hong Kong
852-26668334

TECHNICAL INFORMATION CENTER:

(800) 521-6274

HOME PAGE:

www.motorola.com/semiconductors

MOTOROLA

AN2540/D

Freescale Semiconductor, Inc.