



Intel[®] IXP2XXX Product Line

SRAM Atomic Operations Application Note

August 2004



INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH INTEL® PRODUCTS. EXCEPT AS PROVIDED IN INTEL'S TERMS AND CONDITIONS OF SALE FOR SUCH PRODUCTS, INTEL ASSUMES NO LIABILITY WHATSOEVER, AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY RELATING TO SALE AND/OR USE OF INTEL PRODUCTS, INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT, OR OTHER INTELLECTUAL PROPERTY RIGHT.

Intel Corporation may have patents or pending patent applications, trademarks, copyrights, or other intellectual property rights that relate to the presented subject matter. The furnishing of documents and other materials and information does not provide any license, express or implied, by estoppel or otherwise, to any such patents, trademarks, copyrights, or other intellectual property rights.

Intel products are not intended for use in medical, life saving, life sustaining, critical control or safety systems, or in nuclear facility applications.

Intel may make changes to specifications and product descriptions at any time, without notice.

Designers must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." Intel reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them.

The IXP2800 and IXP2850 Network Processors may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

This SRAM Atomic Operations Application Note as well as the software described in it is furnished under license and may only be used or copied in accordance with the terms of the license. The information in this manual is furnished for informational use only, is subject to change without notice, and should not be construed as a commitment by Intel Corporation. Intel Corporation assumes no responsibility or liability for any errors or inaccuracies that may appear in this document or any software that may be provided in association with this document.

Except as permitted by such license, no part of this document may be reproduced, stored in a retrieval system, or transmitted in any form or by any means without the express written consent of Intel Corporation.

Contact your local Intel sales office or your distributor to obtain the latest specifications and before placing your product order.

Copies of documents which have an ordering number and are referenced in this document, or other Intel literature may be obtained by calling 1-800-548-4725 or by visiting Intel's website at <http://www.intel.com>.

Intel and XScale are registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

*Other names and brands may be claimed as the property of others.

Copyright © 2004, Intel Corporation

Contents

- 1.0 Overview 5
- 2.0 Operation Details 6
- 3.0 Performance 7
 - 3.1 No-Pull Atomics 7
 - 3.2 Address Repetition Rate 8

Figures

- 1 Execution Flow for Atomic test_and_add Operation 6

Tables

- 1 SRAM Atomic Operations 5
- 2 Exclusive Access Guarantees for the Atomic Operation 7
- 3 Capabilities of No-Pull Atomics 7
- 4 Minimum Atomic Repetition Pattern (and Microengine Cycles) for Each Read Latency Case 8

Revision History

Date	Revision	Description
May 2004	001	This is a new document.
August 2004	002	Preparation for web posting.

1.0 Overview

This document provides an overview of the SRAM atomic operations in the IXP2XXX Product Line. First, as background, the micro-architectural details of the SRAM controller are described. Second, performance information helpful to the software developer is detailed.

Atomic operations involve an operand fetch (or pull), an SRAM read (and optional return, or push, of the read data), a data modify operation on the operand and the read data, and an SRAM write of the modify result. The Atomic operations provide exclusive access to a 4-byte SRAM location for the duration of the read-modify-write operation. [Table 1](#) summarizes the SRAM atomic operations.

Table 1. SRAM Atomic Operations

Command	Push Variant	Pull Operand?	Operation
swap	n/a	yes	Swap
set	test_and_set	yes	Multiple bit set (OR)
clr	test_and_clr	yes	Multiple bit clear (AND NOT)
incr	test_and_incr	no	Increment (+ 1)
decr	test_and_decr	no	Decrement (- 1). Saturates at 0.
add	test_and_add	yes	Add. Read data is zero-extended; pull data is sign-extended. Saturates at 0 if pull data most significant bit is 1 (i.e., a negative number).

The SRAM interface uses Quad Data Rate (QDR) devices in Split I/O (SIO) mode. This interface allows for one 4-byte read and one 4-byte write for each SRAM cycle. The IXP2XXX Product Line micro-architecture is designed to have parallel read and write paths to best utilize the QDR bandwidth.

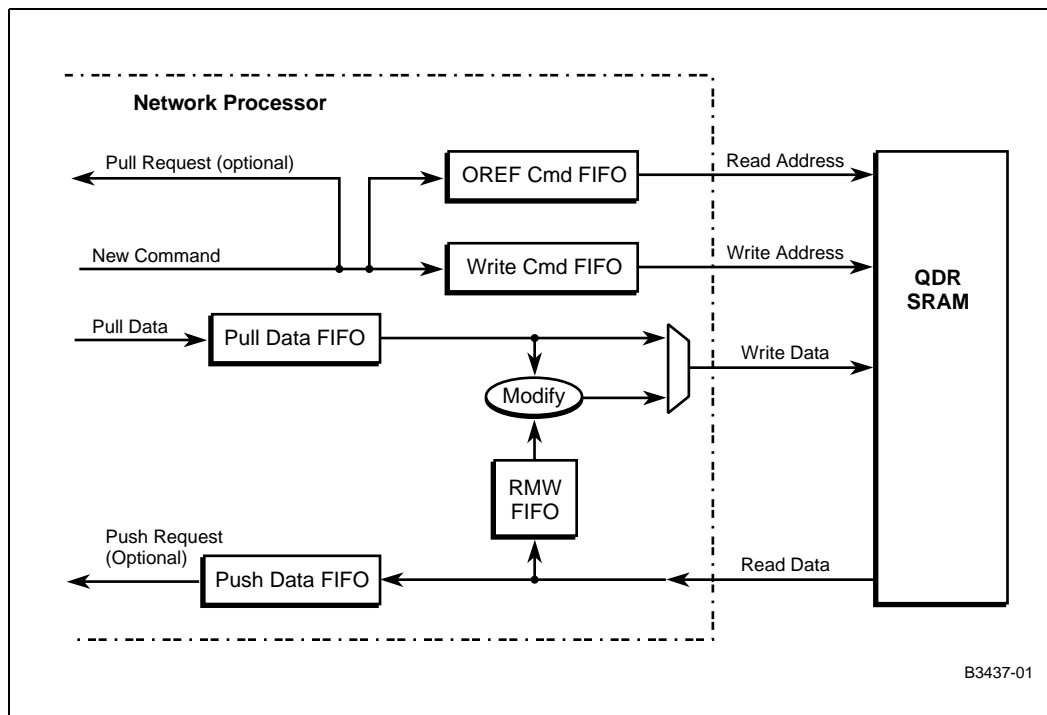
Note: To reach peak performance on atomic operations, *both* the QDR read and write pins must operate at 100 percent bandwidth.

One hazard of separate read and write paths is that commands execute out of order. In fact, this is desirable from a performance point of view. However, for data coherency, reads to a given address must be ordered behind previous writes to the same address. In addition, exclusive access must be guaranteed for atomic operations.

2.0 Operation Details

Figure 1 and the procedural steps that follow, describe the execution flow for an atomic test_and_add operation.

Figure 1. Execution Flow for Atomic test_and_add Operation



The following steps describe the execution flow for an atomic test_and_add operation:

1. The command arrives at the SRAM controller. If necessary, a pull request is issued to get the amount to add to the data in the specified memory location.
2. The command address is compared to the addresses of pending write and atomic commands in the Write Command FIFO. If there is a match, then the command is marked to wait for the preceding command execution. Otherwise, the command is marked as immediately ready. In either case, the test_and_add command is enqueued to the OREF Command FIFO.

Note:

The address compare uses bits [19:6].

3. The test_and_add command is also enqueued to the Write Command FIFO.
4. The read part of the test_and_add command reaches the head of the OREF Command FIFO. If necessary, the command waits at the FIFO head until a pending ordering restriction is met. Then, the read portion of the test_and_add is executed to the QDR SRAM.
5. The SRAM returns the read data. The read data is enqueued to the Push Data FIFO (this is to complete the test portion of the command). In addition, the read data is enqueued to the RMW FIFO.
6. The Modify logic waits for the Pull Data (Pull Data FIFO), the Read Data (RMW FIFO), and the Write Command (Write FIFO). At this point, the add operation is performed and the result is written to SRAM.

Table 2 describes how exclusive access for the atomic operation is guaranteed.

Table 2. Exclusive Access Guarantees for the Atomic Operation

Command	Order Guarantee
Previous Write	The read portion of the Atomic is tagged to wait until all previous write commands to that address are complete.
Previous Atomic	Pending writes for atomic commands are held in the Write Command FIFO, thus the check for previous write commands orders this case as well.
Subsequent Write	The subsequent write command is placed in the Write Command FIFO behind the write for the atomic command. Thus, the Write Command FIFO maintains order and exclusivity.
Subsequent Atomic	The read portion of the second Atomic is tagged to wait behind the Write portion of the current Atomic command.

3.0 Performance

For the IXP2800 Network Processor B-step devices, peak performance for atomic operation is obtained by retiring one atomic command for every SRAM clock cycle. At a 200-MHz SRAM clock, this yields 200 Mega-operations per second. The software or microcode must be written carefully to achieve this peak rate, as described in the following sections.

3.1 No-Pull Atomics

One factor that can stall the completion of an atomic operation is the arrival of Pull Data. For additional information, refer to step 6. in the test_and_add execution flow in Section 2.0. The timing of Pull Data is a function of internal bus usage, and thus is somewhat unpredictable. Therefore, when possible, it is helpful to eliminate the pull altogether, by using no-pull variants of the Atomic commands. In the no-pull variants, the necessary operand is sent along with the command itself on the command_bus.

Note: Inc (increment) and Decr (decrement) never need pull data, so this is not an issue for Inc and Decr commands.

Command_bus size limitations require that only a subset of each atomic command’s function is supported. Table 3 summarizes the capabilities of the no-pull Atomics.

Table 3. Capabilities of No-Pull Atomics

Command	Push Variant	No-Pull Operation
swap	n/a	Swap with 11-bit sign-extended operand
set	test_and_set	Bit set – set a single bit only
clr	test_and_clr	Bit clear – clear a single bit only
add	test_and_add	Add an 11-bit sign-extended operand

Refer to the Intel® IXP2400 and IXP2800 Network Processor Programmer’s Reference Manual for more information about the usage of no-pull atomics.

3.2 Address Repetition Rate

As described in [Section 2.0](#), a new atomic operation will wait to execute if a previous write or atomic command is pending to the same address bits [19:6]. (Atomic commands to non-matching addresses will proceed in parallel.) To achieve best performance on atomics, the rate of arrival for atomic commands *with matching addresses* must be greater than or equal to the time that it takes to execute the first atomic command.

This loop – from the start of execution for the first atomic to the start of execution for the second atomic with a matching address – has two main components: SRAM read latency and controller internal logic latency. The latency in the controller is fixed. Latency is mostly a function of the time required to perform the modify operation and the on-chip delay required to move address/data to and from the IO pads.

SRAM read latency can vary based on board design. For a nominal design (for example, the IXDP2800 advanced development platform, B1 IXP2800), the relevant fields of the SRAM CSRs are set as follows:

```
QDR_INTERNAL_PIPELINE: 1
```

```
SRAM_CONTROL<PIPELINE>: 0
```

This results in a 7-cycle loop for atomic operations. Thus, peak bandwidth is achieved with a repeating stream of 7 (or more) no-pull atomic operations to 7 (or more) locations that all differ in address bits [19:6].

This matches the usage model of employing one SRAM channel to update 7 different statistics in OC-192 minimum packet time (QDR frequency 200 MHz). Peak atomic performance (100% read and write bandwidth) in this usage model has been verified in the B1-step transactor model.

Additional cycles of read latency add to the loop time. Thus, the separation between commands to the same address must be increased, to continue to achieve peak bandwidth.

[Table 4](#) was created by using the B1-step transactor model and varying the read delay via the command `set_sram_channel_pipe_delay` (“chip”, 0, pipeline_delay) where pipeline_delay=0 for the nominal delay described above. The Microengine frequency is 1.4 GHz; QDR frequency is 200 MHz.

[Table 4](#) models a loop of no-pull SRAM atomic operations. For each read latency case, the table shows the minimum atomic repetition pattern, as well as the number of Microengine cycles to execute that pattern. If these guidelines are followed, then 100% QDR bandwidth is obtained for that channel.

Table 4. Minimum Atomic Repetition Pattern (and Microengine Cycles) for Each Read Latency Case

Pipeline_Delay	Number of Atomics	Microengine Cycles	Notes
2	9	63	
1	8	55	approximately an OC-192 data rate
0	7	49	Default case