ANNEX A19: MEMORY PLACEMENT EXTENSIONS

A19.1 INTRODUCTION

This document is an Annex to Volume 1 Release 1.4 of the InfiniBand Architecture, herein referred to as the base document. The base document defines InfiniBand transport and memory registration extensions for supporting memory placement guarantees within the InfiniBand transport. This Annex defines extended placement semantics for applications which require RDMA operations to place data in the responder memory, which can include global visibility and persistence. This Annex is optional in that the implementation of the features described within are optional. If implemented, they must adhere to the compliance statements contained in this Annex.

A19.2 GLOSSARY

Global Visibility
Ensuring the placement of the preceding data accesses in the indicated memory region is visible for reading by the responder platform.

Persistence
Ensuring the placement of the preceding data accesses in the indicated memory region is persistent and the data will be preserved across power cycle or other failure of the responder platform.

A19.3 PROBLEM STATEMENT

As part of a reliable transport service, the base document details delivery which is defined as the guarantee that a message has been delivered to the responder transport only once, in-order and has passed the transport and memory checks required by the transport. The typical use of this reliability guarantee is to release the buffer on the requester side since this packet would no longer need to be retransmitted.
To get further memory placement guarantees, the upper layer protocol libraries typically implement a messaging layer on top of the transport layer using the ordering rules stated in o9-20 of the InfiniBand Base Specification, which specify that an application in the responder platform can depend on the contents of the RDMA WRITE buffer only upon receipt of completion of a subsequent message (RDMA WRITE with IMM or SEND) or an update of memory element by a subsequent atomic operation on the same Queue Pair.

There are upper layer protocols that require further guarantees, for example:

- ULPs that need to ensure that the content of RDMA WRITE buffer is exposed to another queue pair connected to the same responder platform.
- ULPs that need to ensure that the content of RDMA WRITE buffer is globally visible for processes in the responder platform.
- In cases where the underlying memory is persistent, ULPs may need to ensure that the content of the RDMA WRITE buffer is provided with a persistence guarantee.

Using only the placement semantics defined in the base document, such ULPs would need to implement an additional messaging protocol on top of the transport to get the guarantees above, which would require the creation of an ecosystem on top of the transport. This leads to an increase in CPU utilization and latency, and additionally, is not applicable in case the targeted memory is a TCA that does not have any compute resources for processing the ULP messages.
This Annex specifies new transport operations and memory attributes to address the requirements within the transport to enable ULPs to get placement guarantees on RDMA WRITE.

### A19.4 Transport Function Extensions

This section specifies the transport extensions within this Annex, detailing the new operations, the packet formats, ordering rules and error behavior of these operations.

The transport extensions require the memory regions attributes to be extended and have new access permissions for the operations below as described in the memory management section of this paper.

The transport extensions apply for all reliable InfiniBand transports (RC / RD / XRC).

#### A19.4.1 FLUSH Operation

FLUSH is used by the requesting node to achieve guarantees on the data placement within the memory subsystem of preceding accesses to a single memory region, such as those performed by RDMA WRITE, Atomics and ATOMIC WRITE requests.

The operation indicates the virtual address space of a destination node and where the guarantees should apply. This range must be contiguous in the virtual space of the memory key but it is not necessarily a contiguous range of physical memory.

The operation indicates the type of placement guarantee that the requesting node requires as specified in the placement types section below.

**oA19-1:** Responder shall successfully respond on FLUSH operation only after providing the placement guarantees, as specified in the packet, of preceding memory updates (for example: RDMA WRITE, Atomics and ATOMIC WRITE) towards the memory region.

Accesses to memory may be placed at any time. FLUSH can later be used to provide guarantee that the placement has been done.

FLUSH execution may be delayed. Therefore, similar to RDMA READ, it is not strictly ordered to subsequent RDMA WRITEs/Atomics.

FLUSH operation requires memory region level access permission and memory region must provide access permission for FLUSH operation per placement type.
Atomicity and ordering of the placement of preceding accesses (except \texttt{ATOMIC WRITE}) are implementation specific and out of scope of this Annex.

A19.4.1.1 Placement Types

This specification defines two kinds of placement guarantees can be provided by the FLUSH operations:

- Global visibility
- Persistence

FLUSH may require ensuring global visibility, persistence, both or none of them.

A19.4.1.2 Selectivity Level

The scope of addresses being operated on by the FLUSH operation is determined by the selectivity level. Two selectivity levels are defined:

- Memory Region Range
- Memory Region

A FLUSH with a selectivity level of Memory Region Range guarantees that all preceding memory updates in a virtually contiguous range are committed according to the placement type.

A Memory Region selectivity level provides the same guarantees but for an entire R_KEY.

A19.4.1.3 Packet Format

A FLUSH message is built upon FLUSH request packet and is responded successfully by RDMA READ response of zero size.

oA19-2: FLUSH shall be single packet message and shall have no payload.

oA19-3: FLUSH shall be responded by RDMA READ response of zero length only after successful execution of the FLUSH operation according to this specification.

FLUSH packets carry FLUSH extended transport header (Table 1 FLUSH Extended Transport Header on page 5) to specify the placement type and the selectivity level of the operation and RDMA extended header (RETH, see base document RETH definition) to specify the R_Key VA and Length associated with this request following the BTH in RC, RDETH in RD and XRCETH in XRC.
RC FLUSH

| BTH | FETH | RETH |

RD FLUSH

| BTH | RDETH | FETH | RETH |

XRC FLUSH

| BTH | XRCETH | FETH | RETH |

**oA19-4:** FLUSH request shall include FETH followed by RETH immediately following the transport headers of the packet BTH in RC, RDETH in RD and XRCETH in XRC.

**oA19-5:** FLUSH BTH shall hold the Opcode = 0x1C (valid only for RC, RD and XRC, reserved for unreliable transport services).

SE bit shall be set to zero, AckReq shall be ignored, all other field shall be set as specified in the base document.

**oA19-6:** FLUSH BTH header field solicited event (SE) indication shall be set to zero.

**oA19-7:** FLUSH BTH header field AckReq shall be ignored by the responder.

### A19.4.1.3.1 FLUSH EXTENDED TRANSPORT HEADER (FETH)

FLUSH extended transport header (FETH) contains the selectivity level and the memory placement guarantee type.

<table>
<thead>
<tr>
<th>Bytes</th>
<th>Bits</th>
<th>31-6</th>
<th>5-4</th>
<th>3-0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-3</td>
<td>Reserved</td>
<td>Selectivity Level (SEL)</td>
<td>Placement Type (PLT)</td>
<td></td>
</tr>
</tbody>
</table>

**Selectivity Level (SEL)** – defines the memory region scope the FLUSH should apply on. Values are as follows:

- b'00 - Memory Region Range: FLUSH applies for all preceding memory updates to the RETH range on this QP. All RETH fields shall be valid in this selectivity mode. RETH:DMALen field shall be between zero and \(2^{31} - 1\) bytes (inclusive).
• b’01 - Memory Region: FLUSH applies for all preceding memory updates to RETH:R_key on this QP. RETH:DMALen and RETH:VA shall be ignored in this mode.

• b’10 - Reserved.

• b’11 - Reserved.

**Placement Type (PLT)** – Defines the memory placement guarantee of this FLUSH. Multiple bits may be set in this field. Values are as follows:

• Bit 0 if set to ‘1’ indicated that the FLUSH should guarantee Global Visibility.

• Bit 1 if set to ‘1’ indicated that the FLUSH should guarantee Persistence.

• Bits 3:2 are reserved.

**oA19-8**: FLUSH request shall provide placement guarantees according to attributes specified in the FETH, SEL and PLT and RETH header before completing execution in the responder and responding.

### A19.4.1.4 Transport Behavior

FLUSH operation is always a single packet message. It consumes a single PSN and MSN.

**oA19-9**: FLUSH shall consume a single responder resource credit out of the responder resources pool as specified in the base document for RDMA READ and Atomics.

**oA19-10**: Requester QP shall limit the amount of outstanding RDMA READ, Atomics, FLUSH and ATOMIC WRITEs according to the amount of responder resources negotiated in the connection establishment.

**oA19-11**: Responder QP shall have the facilities to implement FLUSH operations as the amount of responder resources negotiated in the establishment of the connection as well as respond to repeated operations.

Upon receiving a FLUSH operation, responder should execute the FLUSH operation according to FETH and RETH fields and respond with RDMA READ Response of length zero and shall be sent only after successful completion of the FLUSH operation.

**oA19-12**: RDMA READ response of length zero shall be sent by the responder in order and only after successful completion of the execution of the FLUSH in the responder.

Access permissions of the memory region targeted by the FLUSH operation shall be verified by the responder before the execution of the FLUSH operation and before execution of any subsequent operation on this QP.
oA19-13: Before executing the FLUSH operation, responder shall verify that the memory region is valid within the provided range described according to SEL and RETH and permitted for operating FLUSH of all placement types required by the operation. In case of mismatch responder shall stop executing the request and all subsequent requests and respond with access violation NAK.

Retry behavior of FLUSH behaves the same as Atomics.Responder shall verify that the repeated packet is valid, is a FLUSH operation, and with matching PSN in the duplicated region as specified in the base document before responding to the repeated packet. Validity of the memory region, as well as access permissions, shall not be checked while executing a repeated operation. Responses on repeated requests must be consistent to the execution of the FLUSH.

oA19-14: FLUSH operation shall be executed only once; duplicated FLUSH packets shall be responded in the same manner as the original response.

oA19-15: For a repeated FLUSH operation responder shall ignore the memory region permission check.

In case the FLUSH operation is not executed on the destination due to error, it is reported to the sender (e.g. an R_Key protection fault) with the
appropriate NAK syndrome response as specified in the base document.
If there is a transport error which results in NAK, then the responder
should not execute any subsequent operations.

A FLUSH operation which has been received and for any reason could not
complete (e.g. memory access error), shall result in a remote operational
error NAK. If an error is detected after the delayed execution such that the
FLUSH operation has not been completed successfully in the memory
subsystem, the responder must not execute any subsequent RDMA
READ and ATOMIC WRITE operations.

\[\text{oA19-16: Upon an operational error of the execution of the FLUSH oper-
ation, the responder shall halt execution of any subsequent ATOMIC
WRITE and RDMA READ operations.}\]

Execution of other subsequent operations (RDMA WRITE/SEND and
Atomics) may have been performed prior to any error detected during
FLUSH execution, according to \[\text{A19.4.2.4 Ordering Rules on page 12}\] of
this document.

Other types of transport errors shall behave as specified in the base doc-
ument. Recovery from access failures such as memory subsystem failure
and unsuccessful completion of FLUSH are implementation specific and
out of the scope of this Annex.

Requester shall notify completion for this operation after receiving RDMA
READ Response of zero size associated with the request by PSN.

**A19.4.2 ATOMIC WRITE OPERATION**

ATOMIC WRITE operation is used by the requesting node to write into the
virtual address space of a destination node ordered to the completion of
preceding Atomic/FLUSH/ATOMIC WRITE operations at the responder
side. ATOMIC WRITE execution updates the underlying memory atomi-
cally and only once.

\[\text{oA19-17: Upon successful receipt of an ATOMIC WRITE operation, re-
ponder must access the memory according to the operation only once,
atomically.}\]

An ATOMIC WRITE operation is built upon ATOMIC WRITE request and
is responded by an RDMA READ response of length zero.

The message size must be 8 bytes and is written to a contiguous range of
the destination QP's virtual address space, access must be aligned to 8
bytes in both the virtual and the physical space.
The responder shall respond to the requester only after the write has been executed toward memory. For atomicity reasons, the underlying physical memory of this operation must be contiguous.

From a visibility standpoint of preceding operations, ATOMIC WRITE plays a similar role as subsequent Atomic, SEND or Write-Immediate on visibility of the content of RDMA WRITE buffer (o9-20 of the base document).

**oA19-18:** An application may depend on the content of an RDMA WRITE buffer at the responder after an update of a memory element by a subsequent ATOMIC WRITE.

### A19.4.2.1 ATOMICITY GUARANTEES

Responder shall ensure the atomicity of committing ATOMIC WRITE towards the responder memory.

Responder applications and concurrent QPs shall either see all the new data of the ATOMIC WRITE or the unmodified previous content.

In the case of subsequent FLUSH operations, ATOMIC WRITE shall be committed to the placement domain type atomically (e.g. for FLUSH with placement type Persistence that has not completed, after power cycle the responder platform should see either the new data or the old data).

### A19.4.2.2 PACKET FORMAT

ATOMIC WRITE is requested by an ATOMIC WRITE request packet and is responded by RDMA READ response of zero size.

**oA19-19:** ATOMIC WRITE shall be single packet message and shall have payload of size 8 bytes.

**oA19-20:** ATOMIC WRITE shall be responded by RDMA READ response of zero length after data after updating the memory.

Following the BTH in RC, DETH in RD and XRCETH in XRC headers ATOMIC WRITE request carries RDMA Extended Header (RETH) to specify the R_Key and VA and Length associated with this request. Length shall be reserved, RETH:VA shall be aligned to 8 bytes.

Following the RETH, ATOMIC WRITE request shall include the 8 byte payload.

**oA19-21:** ATOMIC WRITE requests shall include RETH and payload of 8 bytes immediately following the transport headers of the packet BTH in RC, RDETH in RD / and XRCETH in XRC.
RC ATOMIC WRITE

\[
\begin{array}{c|c|c}
\text{BTH} & \text{FETH} & 8 \text{ Byte Payload} \\
\end{array}
\]

RD ATOMIC WRITE

\[
\begin{array}{c|c|c|c}
\text{BTH} & \text{RDETH} & \text{FETH} & 8 \text{ Byte Payload} \\
\end{array}
\]

XRC ATOMIC WRITE

\[
\begin{array}{c|c|c|c}
\text{BTH} & \text{XRCETH} & \text{FETH} & 8 \text{ Byte Payload} \\
\end{array}
\]

ATOMIC WRITE BTH shall hold the Opcode = 0x1D (valid only for RC, RD and XRC, reserved for unreliable transport services).

SE bit shall be set to zero, AckReq shall be ignored, all other field shall be set as specified in the base document.

- **oA19-22**: ATOMIC WRITE BTH header field solicited event (SE) indication shall be set to zero
- **oA19-23**: ATOMIC WRITE BTH header field AckReq shall be ignored by the responder.

### A19.4.2.3 Transport Behavior

ATOMIC WRITE operation is always a single packet message. It consumes a single PSN and MSN.

- **oA19-24**: ATOMIC WRITE shall consume a single responder resource credit out of the responder resources pool as specified in the base document for RDMA READ and Atomics.

- **oA19-25**: Requester QP shall limit the amount of outstanding RDMA READ, Atomics, FLUSH and ATOMIC WRITEs according to the amount of responder resources negotiated in the connection establishment.

- **oA19-26**: Responder QP shall have the facilities to implement ATOMIC WRITE operations as the amount of responder resources negotiated in the establishment of the connection as well as respond to repeated operations.

Upon receiving ATOMIC WRITE operation, responder should respond with RDMA READ response of length zero after accessing the memory.

- **oA19-27**: RDMA READ response of length zero shall be responded by the responder in order and only after successful execution of the ATOMIC WRITE to responder’s memory.
Responder should validate the targeted memory region has write permission prior to execution, alternatively it should stop executing requests and respond with access violation.

**oA19-28:** Before executing the ATOMIC WRITE operation, the responder shall verify that the memory region is valid and the provided range in RETH is permitted for write access permissions. In case it is not permitted, responder shall stop executing requests and respond with access violation NAK.

![Figure 3 Transport flow for ATOMIC WRITE operation](image)

Retry behavior of ATOMIC WRITE behaves the same as Atomics: Responder shall verify that the repeated packet is valid, is an ATOMIC WRITE operation, and with matching PSN in the duplicated region as specified in the base document before executing a request. Validity of the memory region, as well as access permissions, shall not be checked while executing repeated operation. Responses on repeated requests must be consistent to the execution of the ATOMIC WRITE.

**oA19-29:** ATOMIC WRITE operation shall be executed only once; duplicated ATOMIC WRITE packet shall be responded in the same manner as the original packet.

**oA19-30:** For a repeated ATOMIC WRITE operation, the responder shall ignore the memory region permission check.

Memory key protection of a write access must be checked upon receiving and subsequently executing the operation in the responder platform. If a protection error occurs when receiving the packet, all sub-sequent operations should not be executed, access violation/transport NAK should be sent (i.e. transport checks, headers compliance, etc). In case a protection error occurs when executing the request, Remote Operational NAK...
should be sent to the requester since this error is not precise (subsequent operations may have been executed).

Requester shall notify completion for this operation after receiving RDMA READ Response of zero size associated with the request by PSN.

**A19.4.2.4 ORDERING RULES**

This section is summarizing the ordering of execution of a responder QP within an HCA as defined in the base document along with the extensions defined in this Annex.

FLUSH and ATOMIC WRITE are designed to be delayed operations, due to their high overhead with the memory subsystem, nevertheless the two operations are ordered with respect to one another. Specifically, in the ordering domain of delayed operations, these operations are ordered.

**oA19-31:** An inbound ATOMIC WRITE / FLUSH Request shall not begin execution until execution of ALL previous received inbound Requests has begun.

Inbound ATOMIC WRITE / FLUSH request execution may be delayed. Therefore, subsequent inbound RDMA WRITE, SEND and Atomic Requests may bypass execution of ATOMIC WRITE / FLUSH.

**oA19-32:** Within a QP, Inbound ATOMIC WRITE / FLUSH shall not be executed towards memory until all preceding operations, except RDMA READ, have been completed by the CA.

Inbound ATOMIC WRITE / FLUSH may be executed before or after preceding RDMA READ (due to its repeated behavior) and shall be executed before any subsequent RDMA READ. Requester may use a fence to enforce any required ordering.

Responder sends RDMA READ response of zero length in order after the operation has been completed.

**oA19-33:** Within a QP, Responder shall transmit responses in order as specified in the base document, including responses specified in this Annex.

All ordering rules above are in the scope of a single QP.
This table below summarizes the responder execution ordering rules of the transport as defined in the spec and as extended in this Annex.

### Table 2: Responder execution ordering rules

<table>
<thead>
<tr>
<th>FIRST</th>
<th>SECOND</th>
<th>RDMA WRITE/SEND</th>
<th>Atomics</th>
<th>RDMA READ</th>
<th>FLUSH/ATOMIC WRITE</th>
</tr>
</thead>
<tbody>
<tr>
<td>RDMA WRITE/SEND</td>
<td>#</td>
<td>#</td>
<td></td>
<td>#</td>
<td>#</td>
</tr>
<tr>
<td>Atomics</td>
<td>#</td>
<td>#</td>
<td></td>
<td>#</td>
<td>#</td>
</tr>
<tr>
<td>RDMA READ</td>
<td>F</td>
<td>F</td>
<td></td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>FLUSH/ATOMIC WRITE</td>
<td>F</td>
<td>F</td>
<td></td>
<td>#</td>
<td>#</td>
</tr>
</tbody>
</table>

### Table 3: Responder execution Key

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>#</td>
<td>In Order</td>
</tr>
<tr>
<td>F</td>
<td>Ordering requires requester fence.</td>
</tr>
</tbody>
</table>

**Note:** Ordering is per QP
An example of an application that could utilize the ordering rules defined here is an application that would like to flush data access to persistency and then, after data is already persistent on the responder platform, update atomically a pointer in persistent memory. This can be achieved by posting a FLUSH request to the data followed by an ATOMIC WRITE request of the pointer. The pointer will be updated in persistent memory only after data is committed to persistent memory as illustrated in the figure below.

![Diagram showing ATOMIC WRITE ordering rules example](image)

**Figure 4  ATOMIC WRITE ordering rules example**

In the example above, FLUSH is followed by ATOMIC WRITE, ATOMIC WRITE execution is postponed until FLUSH operation has been completed in the responder platform.

**A19.4.3 Software Transport Interface**

This section will describe the software transport interface changes driven by this Annex. Chapters that are unaffected are omitted from this Annex (for example: resources states).

**A19.4.3.1 HCA Resources**

This Annex introduces the following new HCA attributes:

- Ability to support Memory Placement Extensions
a) Ability to support FLUSH
   i) Ability to support FLUSH with PLT Global Visibility
   ii) Ability to support FLUSH with PLT Persistence
b) Ability to support ATOMIC WRITE

### A19.4.3.2 Queue Pair

This specification introduces the following new QP attributes in case the HCA supports memory placement extensions.

- QP should have an additional optional attribute of enabled/disabled for FLUSH.
  - Change of this attribute is possible on all QP transition types that enable/disable RDMA. Atomics can be changed as described in table 96 of the InfiniBand Base Specification (e.g. RTS to RTS).

- QP Errors:
  - Similar to RDMA READ, FLUSH operation may be partially completed, unless FLUSH completion was reported which indicates the operation has been fully completed.
  - Similar to Atomics, ATOMIC WRITE may have been executed or not, execution of ATOMIC WRITE shall not be partial.

### A19.4.3.3 Memory Management

In case the memory extension is supported by the CA, in addition to permissions in the base document Chapter 10 Section 10.6.3.2, the consumer should specify, on a memory region basis, the required permissions for remote FLUSH access of the region in the base document Chapter 10 Section 10.6.7.2. There are no local permissions for FLUSH operation.

If the memory region is permitted for FLUSH, the region should also specify permissions for the variety of placement types (PLT). This should be a multi-bit field as several PLTs could be supported by a single region.

The following PLTs are specified by this Annex:

- Global Visibility
- Persistence

FLUSH shall be permitted by the CA only if its R_Key is associated with memory region that enables FLUSH and enables the placement types described in the request. Alternatively, this operation shall be considered as access violation.

ATOMIC WRITE shall be permitted by the CA only if the memory region is enabled for remote write operations.
oA19-34: ATOMIC WRITE operation which is targeted to an R_Key without remote write permissions shall be considered as an access violation.

Similar to RDMA READ (o10-37.2.21), succeeding Send with Invalidate, invalidation may occur before all preceding ATOMIC WRITE and FLUSH have been completed.

A19.4.3.4 WORK REQUESTS

In case memory extensions are supported by the CA, the CA should support the following new work requests as part of the RDMA work request, FLUSH and ATOMIC WRITE. The new operations, new WQE modifier and transactions ordering rules are described in this Annex.

Memory placement extension apply for reliable transports as specified in Table 4.

<table>
<thead>
<tr>
<th>Operational Type Matrix</th>
</tr>
</thead>
<tbody>
<tr>
<td>FLUSH</td>
</tr>
<tr>
<td>RC</td>
</tr>
<tr>
<td>RD</td>
</tr>
<tr>
<td>UC</td>
</tr>
<tr>
<td>UD</td>
</tr>
<tr>
<td>RAW</td>
</tr>
<tr>
<td>XRC Initiator</td>
</tr>
</tbody>
</table>

A19.4.4 FLUSH

FLUSH operation is used by the requesting node to achieve guarantees on the data placement within the memory subsystem of previously posted RDMA WRITE, Atomics and ATOMIC WRITE messages.

FLUSH is posted to a send queue.

The operation indicates the virtual address space of the R_Key in the destination node where the guarantees should apply (not necessarily a contiguous range of physical memory) and Selectivity level (SEL) to indicate whether the request applies to the entire R_Key or specific region in the R_Key.

In case FLUSH with selectivity level of memory region range, operation shall indicate the remote address, R_Key and the length of the region that the operation applies on. In case of FLUSH with selectivity level of memory region, operation shall indicate the R_Key.
The operation indicates types of memory placement types (PLT) that the requesting node requests to guarantee, e.g. global visibility and persistence.

The operation is supported only in reliable transports.

FLUSH requests made to the Send Queue are guaranteed to be serialized after all RDMA WRITEs, Atomics and ATOMIC WRITEs posted on the Send Queue.

FLUSH operations are posted to the Send Queue using the Post Send Request Verb. The completion for this request is indicated only if the FLUSH operation was successful and previous operations were placed in memory according the type.

Error flows are described in the transport section of this Annex, recovery from FLUSH completion with error is out of the scope of this specification.

This operation requires new Post Send operation type; work request modifiers are the same as RDMA WRITE with additional PLT and SEL

A19.4.5 ATOMIC WRITE

The ATOMIC WRITE operation is used by the application to write into the virtual address space of a destination node ordered to the completion of preceding Atomic/FLUSH/ATOMIC WRITE.

Execution of the write should update the underlying memory atomically and only once. The message size is 8 bytes and is written to a contiguous range of the destination QP's virtual address space.

Access must be aligned to 8 bytes in both the virtual and the physical space.

The operation is supported only in reliable transports. If the CA supports this operation, it should support it across all reliable transports.

This operation requires new Post Send operation type; work request modifiers are specified in Table 5 Work Request Modifiers Matrix on page 18.
## A19.4.6 WORK REQUEST MODIFIERS MATRIX

<table>
<thead>
<tr>
<th></th>
<th>ATOMIC WRITE</th>
<th>FLUSH</th>
</tr>
</thead>
<tbody>
<tr>
<td>Work Request ID</td>
<td>Required</td>
<td>Required</td>
</tr>
<tr>
<td>Completion notification indicator</td>
<td>Required if Send Queue Signaling Type is Selectable</td>
<td>Required if Send Queue Signaling Type is Selectable</td>
</tr>
<tr>
<td>Scatter/Gather list</td>
<td>Required</td>
<td>N/A</td>
</tr>
<tr>
<td># of Data Segments</td>
<td>Required</td>
<td>N/A</td>
</tr>
<tr>
<td>Immediate Data</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Fence Indicator</td>
<td>Optional</td>
<td>Optional</td>
</tr>
<tr>
<td>Remote Node Address</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Remote Node QP # and Q_Key</td>
<td>Required for Reliable Datagram QPs</td>
<td>Required for Reliable Datagram QPs</td>
</tr>
<tr>
<td>EE Context</td>
<td>Required for Reliable Datagram QPs</td>
<td>Required for Reliable Datagram QPs</td>
</tr>
<tr>
<td>Remote address</td>
<td>Required</td>
<td>Required if SEL is Memory Region Range</td>
</tr>
<tr>
<td>Remote R_Key</td>
<td>Required</td>
<td>Required</td>
</tr>
<tr>
<td>Atomic operands</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>FLUSH Operands (SEL &amp; PLT)</td>
<td>N/A</td>
<td>Required</td>
</tr>
<tr>
<td>Solicited Event</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Ethertype</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Maximum Static Rate</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Local Invalidate Fence</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>New key to use on L_KEY and R_KEY</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>MR Handle</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>MW Handle</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>PBL</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>First Byte Offset</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Addressing Type</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>VA</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>IOVA</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>_Key</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Local R_KEY</td>
<td>N/A</td>
<td>N/A</td>
</tr>
</tbody>
</table>
### Table 5  WORK REQUEST MODIFIERS MATRIX

<table>
<thead>
<tr>
<th></th>
<th>ATOMIC WRITE</th>
<th>FLUSH</th>
</tr>
</thead>
<tbody>
<tr>
<td>Access Control</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Length</td>
<td>N/A</td>
<td>Required if SEL is Memory Region Range</td>
</tr>
<tr>
<td>Remote XRC SRQ Number</td>
<td>Required</td>
<td>Required</td>
</tr>
</tbody>
</table>

#### A19.4.7 TRANSACTION ORDERING RULES

The table below is referenced in the base document (Chapter 10.8.3.3).

### Table 6  Work Request Operation Ordering

<table>
<thead>
<tr>
<th>FIRST</th>
<th>SECOND</th>
<th>Send</th>
<th>Bind Window</th>
<th>WRITE</th>
<th>READ</th>
<th>Atomic</th>
<th>Fast Register</th>
<th>Local Invalidate</th>
<th>ATOMIC WRITE</th>
<th>FLUSH</th>
</tr>
</thead>
<tbody>
<tr>
<td>SEND</td>
<td>#</td>
<td>#</td>
<td>#</td>
<td>#</td>
<td>#</td>
<td>#</td>
<td>NR</td>
<td>L</td>
<td>#</td>
<td>#</td>
</tr>
<tr>
<td>Bind Window</td>
<td>#</td>
<td>#</td>
<td>#</td>
<td>#</td>
<td>#</td>
<td>#</td>
<td>NR</td>
<td>L</td>
<td>#</td>
<td>#</td>
</tr>
<tr>
<td>WRITE</td>
<td>#</td>
<td>#</td>
<td>#</td>
<td>#</td>
<td>#</td>
<td>#</td>
<td>NR</td>
<td>L</td>
<td>#</td>
<td>#</td>
</tr>
<tr>
<td>READ</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>#</td>
<td>F</td>
<td>NR</td>
<td>L</td>
<td>F</td>
<td>#</td>
</tr>
<tr>
<td>Atomic</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>#</td>
<td>F</td>
<td>NR</td>
<td>L</td>
<td>F</td>
<td>#</td>
</tr>
<tr>
<td>Fast Register</td>
<td>#</td>
<td>#</td>
<td>#</td>
<td>#</td>
<td>#</td>
<td>#</td>
<td>#</td>
<td></td>
<td>L</td>
<td>#</td>
</tr>
<tr>
<td>Local Invalidate</td>
<td>#</td>
<td>#</td>
<td>#</td>
<td>#</td>
<td>#</td>
<td>#</td>
<td>#</td>
<td></td>
<td>#</td>
<td>#</td>
</tr>
<tr>
<td>ATOMIC WRITE</td>
<td>#</td>
<td>#</td>
<td>#</td>
<td>#</td>
<td>#</td>
<td>#</td>
<td>NR</td>
<td>L</td>
<td>#</td>
<td>#</td>
</tr>
<tr>
<td>FLUSH</td>
<td>#</td>
<td>#</td>
<td>#</td>
<td>#</td>
<td>#</td>
<td>#</td>
<td>NR</td>
<td>L</td>
<td>#</td>
<td>#</td>
</tr>
</tbody>
</table>

### Table 7  Ordering Rules Key

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>#</td>
<td>Order is always maintained.</td>
</tr>
<tr>
<td>NR</td>
<td>Order is not required to be maintained between the Fast Register and the previous operations.</td>
</tr>
<tr>
<td>F</td>
<td>Order maintained only if second operation has Fence Indicator set.</td>
</tr>
<tr>
<td>L</td>
<td>Order maintained only if Invalidate operation has Local Invalidate Fence Indicator set</td>
</tr>
</tbody>
</table>