I/O Design Architecture (IODA3) Compliance
Test Harness and Test Suite (TH/TS)
Workgroup Specification
Revision 1.0 (June 23, 2020)
Abstract

The purpose of the OpenPOWER I/O Design Architecture, version 3 (IODA3) Compliance Test Harness and Test Suite (TH/TS) Specification is to provide the test suite requirements to be able to demonstrate OpenPOWER I/O Design Architecture, version 3 (IODA3) compliance for POWER9™ systems. It describes the tests required in the test suite and a test harness needed to execute the test suite. It also describes the successful execution of the test suite, including what it means for an optional feature to fail. The input to this specification is the OpenPOWER I/O Design Architecture, version 3 (IODA3) Specification which describes the chip architecture for key aspects of PCI Express® (PCIe)-based host bridge (PHB) designs for POWER9 systems. This specification defines the PHB hardware and firmware requirements for these functions: 1. MMIO Partitionable-Endpoint Number Determination, 2. DMA and MSI Partitionable-Endpoint Number Determination, 3. Partitionable-Endpoint State and Enhanced Error Handling, 4. Error-Injection, 5. DMA with No Page Migration, 6. DMA with Page Migration, 7. DMA with Multilevel TCE Tables, 8. DMA Read Sync Register, 9. Message-Signalled Interrupt, 10. PCIe Configuration Space, and 11. Partitionable-Endpoint State Table.

This document is a Standard Track, Work Group Specification work product owned by the Compliance Workgroup and handled in compliance with the requirements outlined in the OpenPOWER
Foundation Work Group (WG) Process document. It was created using the Master Template Guide version 1.0.0. Comments, questions, etc. can be submitted to the public mailing list for this document at <openpower-ioda-thts@mailinglist.openpowerfoundation.org>.
# Table of Contents

Preface ........................................................................................................................................... v
  1. Conventions ....................................................................................................................... v
  2. Document change history ............................................................................................ vi
1. Introduction ................................................................................................................................ 1
  1.1. Conformance to this Specification ................................................................................... 2
2. IODA3 Test Harness .................................................................................................................. 5
3. IODA3 Test Suite ....................................................................................................................... 6
  3.1. MMIO Partitionable-Endpoint Number Determination ....................................................... 6
  3.2. DMA and MSI Partitionable-Endpoint Number Determination ............................................ 7
  3.3. Partitionable-Endpoint State and Enhanced Error Handling ............................................. 9
  3.4. Error-Injection ................................................................................................................ 10
  3.5. DMA with No Page Migration ........................................................................................ 11
  3.6. DMA with Page Migration ............................................................................................. 12
  3.7. DMA with Multilevel TCE Tables ................................................................................... 14
  3.8. DMA Read Sync Register ............................................................................................. 15
  3.9. Message-Signalled Interrupt .......................................................................................... 16
  3.10. PCIe Configuration Space ........................................................................................... 18
  3.11. Partitionable-Endpoint State Table ............................................................................... 18
  3.12. Successful Execution of the Required Tests ................................................................ 19
  3.13. Successful Execution of the Optional Tests ................................................................ 19
A. Glossary .................................................................................................................................. 20
B. OpenPOWER Foundation overview ......................................................................................... 21
  B.1. Foundation documentation ............................................................................................ 21
  B.2. Technical resources ...................................................................................................... 21
  B.3. Contact the foundation .................................................................................................. 22
Preface

1. Conventions

The OpenPOWER Foundation documentation uses several typesetting conventions.

Notices

Notices take these forms:

- **Note**
  
  A handy tip or reminder.

- **Important**
  
  Something you must be aware of before proceeding.

- **Warning**
  
  Critical information about the risk of data loss or security issues.

Changes

At certain points in the document lifecycle, knowing what changed in a document is important. In these situations, the following conventions will be used.

- **New text will appear like this.** Text marked in this way is completely new.

- **Deleted text will appear like this.** Text marked in this way was removed from the previous version and will not appear in the final, published document.

- **Changed text will appear like this.** Text marked in this way appeared in previous versions but has been modified.

Command prompts

In general, examples use commands from the Linux operating system. Many of these are also common with Mac OS, but may differ greatly from the Windows operating system equivalents.

For the Linux-based commands referenced, the following conventions will be followed:

- **$ prompt** Any user, including the root user, can run commands that are prefixed with the $ prompt.

- **# prompt** The root user must run commands that are prefixed with the # prompt. You can also prefix these commands with the sudo command, if available, to run them.
Document links

Document links frequently appear throughout the documents. Generally, these links include a text for the link, followed by a page number in parenthesis. For example, this link, Preface [v], references the Preface chapter on page v.

2. Document change history

This version of the guide replaces and obsoletes all earlier versions.

The following table describes the most recent changes:

<table>
<thead>
<tr>
<th>Revision Date</th>
<th>Summary of Changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>June 23, 2020</td>
<td>• Rev 1.0 - OpenPOWER IODA3 Compliance Test Harness and Test Suite - Workgroup Approved Specification</td>
</tr>
<tr>
<td>March 25, 2020</td>
<td>• Rev 1.0-PRD - OpenPOWER IODA3 Compliance Test Harness and Test Suite - Workgroup Approved Public Review Draft</td>
</tr>
<tr>
<td>March 17, 2020</td>
<td>• Rev 1.0-pre4 - added glossary to define the acronyms as discussed at the March 17, 2020 Compliance Work Group meeting</td>
</tr>
<tr>
<td>March 6, 2020</td>
<td>• Rev 1.0-pre3 - updated Abstract, defined acronyms on first use, and other minor changes</td>
</tr>
<tr>
<td>February 17, 2020</td>
<td>• Rev 1.0-pre2 - fixed typos and changed formatting</td>
</tr>
<tr>
<td>February 14, 2020</td>
<td>• Rev 1.0-pre1 - initial draft</td>
</tr>
</tbody>
</table>
1. Introduction

The purpose of the OpenPOWER I/O Design Architecture, version 3 (IODA3) Compliance Test Harness and Test Suite (TH/TS) Specification is to provide the test suite requirements to be able to demonstrate OpenPOWER I/O Design Architecture, version 3 (IODA3) compliance for POWER9 systems. It contains the following:

• Section describing the test harness needed to execute the test suite
• Section describing the tests required to be in the test suite
• Section describing the successful execution of the test suite, including what it means for an optional feature to fail

The input to this specification is the following specification:

1. OpenPOWER I/O Design Architecture, version 3 (IODA3) Specification which describes the chip architecture for key aspects of PCI Express® (PCIe)-based host bridge (PHB) designs for POWER9™ systems.

This specification defines the PHB hardware and firmware requirements for the functions shown in the following diagram.
1.1. Conformance to this Specification

The following lists a set of numbered conformance clauses to which any implementation of this specification must adhere in order to claim conformance to this specification (or any optional portion thereof):


2. For optional facilities that are implemented, the optional tests in the Section 3.1.2, “MMIO Partitionable-Endpoint Number Determination Test Suite Optional Tests” [7] must be successfully executed.


4. For optional facilities that are implemented, the optional tests in the Section 3.2.2, “DMA and MSI Partitionable-Endpoint Number Determination Test Suite Optional Tests” [8] must be successfully executed.

5. The required tests in the Section 3.3.1, “Partitionable-Endpoint State and Enhanced Error Handling Test Suite Required Tests” [9] must be successfully executed.

6. For optional facilities that are implemented, the optional tests in the Section 3.3.2, “Partitionable-Endpoint State and Enhanced Error Handling Test Suite Optional Tests” [10] must be successfully executed.

   Note

   There are no Partitionable-Endpoint State and Enhanced Error Handling test suite optional tests.


8. For optional facilities that are implemented, the optional tests in the Section 3.4.2, “Error-Injection Test Suite Optional Tests” [11] must be successfully executed.

   Note

   There are no Error-Injection test suite optional tests.


10. For optional facilities that are implemented, the optional tests in the Section 3.5.2, “DMA with No Page Migration Test Suite Optional Tests” [12] must be successfully executed.

   Note

   There are no DMA with No Page Migration test suite optional tests.

12. For optional facilities that are implemented, the optional tests in the Section 3.6.2, “DMA with Page Migration Test Suite Optional Tests” [14] must be successfully executed.

**Note**

There are no DMA with Page Migration test suite optional tests.


14. For optional facilities that are implemented, the optional tests in the Section 3.7.2, “DMA with Multilevel TCE Tables Test Suite Optional Tests” [15] must be successfully executed.

**Note**

There are no DMA with Multilevel TCE Tables test suite optional tests.

15. The required tests in the Section 3.8.1, “DMA Read Sync Register Test Suite Required Tests” [16] must be successfully executed.

16. For optional facilities that are implemented, the optional tests in the Section 3.8.2, “DMA Read Sync Register Test Suite Optional Tests” [16] must be successfully executed.

**Note**

There are no DMA Read Sync Register test suite optional tests.


18. For optional facilities that are implemented, the optional tests in the Section 3.9.2, “Message-Signalled Interrupt Test Suite Optional Tests” [17] must be successfully executed.

**Note**

There are no Message-Signalled Interrupt test suite optional tests.


20. For optional facilities that are implemented, the optional tests in the Section 3.10.2, “PCIe Configuration Space Test Suite Optional Tests” [18] must be successfully executed.

**Note**

There are no PCIe Configuration Space test suite optional tests.

21. The required tests in the Section 3.11.1, “Partitionable-Endpoint State Table Test Suite Required Tests” [18] must be successfully executed.
22. For optional facilities that are implemented, the optional tests in the **Section 3.11.2, “Partitionable-Endpoint State Table Test Suite Optional Tests”** [19] must be successfully executed.

**Note**

There are no Partitionable-Endpoint State Table test suite optional tests.
2. IODA3 Test Harness

The IODA3 Test Harness is an OpenPOWER Ready system that has I/O Adapters with Linux and firmware code loaded. The purpose of the Test Harness is to provide an environment to test for the existence and correct behavior of required IODA3 facilities and implemented optional IODA3 facilities.
3. IODA3 Test Suite

The purpose of this chapter is to provide the test suite requirements to be able to demonstrate OpenPOWER IODA3 compliance. The Template Test Harness or any other test harness should test the following functions.

1. MMIO Partitionable-Endpoint Number Determination
2. DMA and MSI Partitionable-Endpoint Number Determination
3. Partitionable-Endpoint State and Enhanced Error Handling
4. Error-Injection
5. DMA with No Page Migration
6. DMA with Page Migration
7. DMA with Multilevel TCE Tables
8. DMA Read Sync Register
9. Message-Signalled Interrupt
10. PCIe Configuration Space
11. Partitionable-Endpoint State Table

Details on the required test and optional tests are in the following sections.

3.1. MMIO Partitionable-Endpoint Number Determination

The MMIO Partitionable-Endpoint Number Determination tests ensure that the PCI host bridge hardware and firmware supports Base Address Register (BAR) spaces of the devices below the PCI host bridge, MMIO address-space decoding, and the assignment of partitionable-endpoint numbers to those decodes.

3.1.1. MMIO Partitionable-Endpoint Number Determination Test Suite Required Tests

The MMIO Partitionable-Endpoint Number Determination required tests need to check all of the following required functions and verify correct behavior.

1. PCI host bridge hardware supports the decoding of MMIO addresses.

2. PCI host bridge hardware provides enough address-space decodes to support the necessary, probably noncontiguous, Base Address Register (BAR) spaces of the devices to be located below the PCI host bridge, including legacy devices that require an address programmed in their BARs that are below 4 GB.

3. PCI host bridge hardware provides capability for the address decodes to be assigned an appropriate partitionable-endpoint number.

4. The platform firmware sets up any chip implementation-specific address ranges appropriately.
3.1.2. MMIO Partitionable-Endpoint Number Determination Test Suite Optional Tests

The MMIO Partitionable-Endpoint Number Determination optional tests need to check all of the implemented optional functions and verify correct behavior. Optional functions are shown below.

1. When multiple address decodes in the PCI host bridge hardware are provided for any given function, the partitionable-endpoint number assigned is the same.

3.2. DMA and MSI Partitionable-Endpoint Number Determination

The DMA and MSI Partitionable-Endpoint Number Determination tests ensure that the PCI host bridge hardware and firmware supports DMA and MSI address-space decoding and the assignment of partitionable-endpoint numbers to those decodes, including requester ID translation table (RTT) and partitionable-endpoint lookup table-vector (PELT-V) tables in system memory with RTT BAR and PELT-V BAR, caching and invalidating cached requester ID translation entries (RTEs) in the requester ID translation cache (RTC), performing requester ID (RID) translation, and recognizing and interrupting for an invalid RID.

3.2.1. DMA and MSI Partitionable-Endpoint Number Determination Test Suite Required Tests

The DMA and MSI Partitionable-Endpoint Number Determination required tests need to check all of the following required functions and verify correct behavior.

1. PCI host bridge hardware implements the RTT in system memory with the entries defined by RTE Definition table of the IODA3 Specification and provides a register that firmware can set to point to the starting address of that table.

2. PCI host bridge hardware implements a BAR to point to the start of the RTT (RTT BAR), loadable by the firmware.

3. PCI host bridge hardware implements the PELT-V in system memory with entries defined by PELE-V Definition table of the IODA3 Specification and provides a register that firmware can set to point to the starting address of that table.

4. PCI host bridge hardware implements a BAR to point to the start of the PELT-V (PELT-V BAR), loadable by the firmware.

5. PCI host bridge hardware provides an RID translation cache (RTC) for caching RTEs used for DMA operations.

6. PCI host bridge hardware provides an RTC Invalidate Register, as defined by RTC Invalidate Register Definition table of the IODA3 Specification, for invalidating individual cached RTEs or all RTEs. The hardware stops using the entry when firmware indicates to invalidate, but it can wait until the RTC entry is used once by a DMA operation. A Store to this register performs the specified invalidation operation. A Load from this register returns the last value stored to this register.
7. During RID translation, if the RID is in the RTC, the PCI host bridge hardware uses the cached value.

8. During RID translation, if the RID is not in the RTC, the PCI host bridge hardware uses the RID as an index into the RTT and reads the RTE. If this is a DMA operation, cache the entry. For error messages, if the partitionable-endpoint number is not all ones, use the PE# field in the RTE as a PELT-V index. Access the PELT-V entry and use the bit array obtained as the array of partitionable-endpoint numbers that are affected by the error.

9. When accessing the RTT, if the RTE is all ones, an invalid RID has been received. The hardware sets the appropriate error bit in the PCI host bridge, stores the RID information, and interrupts the firmware for processing of the error.

10. The platform firmware sets up the RTT BAR and PELT-V BAR to point to the start of those structures in contiguous real system memory, with a size that is a power of 2 and with an address alignment on an integer multiple of the size of the table.

11. To change the RTT BAR or PELT-V BAR while DMA operations might be in progress, the firmware first creates the new table and changes the BAR. Then, firmware makes sure that all DMA operations that are queued in the PCI host bridge (DMA write/read and MSIs) are completed before reusing the system memory locations that were previously used by the table.

12. For RIDs that do DMA operations or that issue MSIs, the platform firmware sets up the partitionable-endpoint number in the RTT to the partitionable-endpoint number that is associated with the RID. There might be more than one RID associated with the same partitionable-endpoint number.

13. For each partitionable-endpoint number, the platform firmware creates an entry in the PELT-V with an offset equal to the partitionable-endpoint number. That entry contains the appropriate bits set for each partitionable endpoint that might be affected by an error against the RID associated with the partitionable-endpoint number.

14. For invalid RIDs (that is, ones that are not configured in the PCIe hierarchy), the platform firmware sets the RTE to all ones.

15. The platform firmware manages the RTT and PELT-V entries in system memory and the RTC on the PCI host bridge chip appropriately at all times, including during hot plug and dynamic logical partitioning (DLPAR) operations.

3.2.2. DMA and MSI Partitionable-Endpoint Number Determination Test Suite Optional Tests

The DMA and MSI Partitionable-Endpoint Number Determination optional tests need to check all of the implemented optional functions and verify correct behavior. Optional functions are shown below.

1. PCI host bridge hardware provides an RID translation cache (RTC) for caching RTEs used for MSI operations and for error message operations.

2. For MSI operations and for error messages, the PCI host bridge hardware caches the entry, when during RID translation the RID is not in the RTC and the hardware reads the RTE using the RID as an index into the RTT in system memory.
3.3. Partitionable-Endpoint State and Enhanced Error Handling

The Partitionable-Endpoint State and Enhanced Error Handling tests ensure that the PCI host bridge hardware and firmware supports the Enhanced Error Handling function. This includes the capability of Enhanced-Error-Handling Enablement state, the MMIO Stopped state, and DMA Stopped state, detecting failures and entering the appropriate state, responding appropriately when in one of these states, and resetting the state. All of this capability is provided for each partitionable endpoint.

3.3.1. Partitionable-Endpoint State and Enhanced Error Handling Test Suite Required Tests

The Partitionable-Endpoint State and Enhanced Error Handling required tests need to check all of the following required functions and verify correct behavior.

1. Each partitionable endpoint’s MMIO Stopped state and DMA Stopped state are independent of each other. The hardware gives the firmware a way to set and clear the DMA Stopped state and the MMIO Stopped state:
   - Independently from each other
   - Independent for those Stopped states for other partitionable endpoints
   - Atomically with any other errors that might be occurring at the time

2. For any detected failure to or from a partitionable endpoint, the PCI host bridge hardware sets both the MMIO Stopped and DMA Stopped states for the partitionable endpoint.

   **Note**

   Exception: Not required if the error that caused the failure can be reported to the I/O adapter function in a way that enables it to report the error to its device driver while avoiding any data corruption.

3. If an I/O fabric consists of a hierarchy of components, when a failure is detected in the fabric and that failure cannot be isolated to a single partitionable endpoint, the PCI host bridge hardware puts all partitionable endpoints that are downstream of the failure into the MMIO Stopped and DMA Stopped states if they might be affected by the failure.

4. From the time that the MMIO Stopped state is entered for a partitionable endpoint, the PCI host bridge hardware prevents the partitionable endpoint from responding to Load and Store operations including the operation that caused the partitionable endpoint to enter the MMIO Stopped state. A Load operation returns all ones with no error indication. A Store operation is discarded until the firmware directs the hardware otherwise or until the PCI host bridge chip is reset. That is, Load and Store operations are treated as if they received a conventional PCI master abort error.

5. From the time that the DMA Stopped state is entered for a partitionable endpoint, the PCI host bridge hardware prevents the partitionable endpoint from initiating a new DMA request or completing a DMA request that caused the partitionable endpoint to enter the DMA Stopped state, including MSI or MSI-X DMA operations, until the firmware directs the hardware otherwise or until the PCI host bridge chip is reset. DMA requests that were started before the DMA
Stopped state is entered can be completed. DMA requests requiring a response that are discard-
ded due to the partitionable endpoint being in the DMA Stopped state (for example, a read
request or an atomic request), return an unsupported request (UR) to the requester. DMA read
response data that is returned to the PCI host bridge after the setting of the DMA Stopped
state is returned to the requester with a Completer Abort status, if possible. Otherwise, discard
the response; for example, a link down condition is a case where return of a response is not
possible.

6. The PCI host bridge hardware provides the capability to the firmware to determine, on a per-
partitionable-endpoint basis, that a failure occurred which caused the partitionable endpoint to
be put into the MMIO Stopped and DMA Stopped states and to read the actual state information
(MMIO Stopped state and DMA Stopped state).

7. The PCI host bridge hardware provides the capability of separately enabling and resetting the
DMA Stopped and MMIO Stopped states for a partitionable endpoint without disturbing other
partitionable endpoints on the platform. The hardware provides this capability without requir-
ing a partitionable-endpoint reset and does so through normal processor Store instructions.
Firmware enabling of MMIO or DMA Stopped states has the same, immediate effect as if a PCI
host bridge-detected error set those states.

8. The PCI host bridge hardware provides the capability to the firmware to deactivate all provided
resets (hot reset, fundamental reset), independent of other resets. The hardware provides the
proper controls on the reset transitions to prevent failures from being introduced into the platform
by changing the reset.

9. The PCI host bridge hardware provides the capability to the firmware to activate all provided
resets (hot reset, fundamental reset), independent of other resets. The hardware provides the
proper controls on the reset transitions to prevent failures from being introduced into the platform
by changing the reset.

10. When a partitionable endpoint is put into the MMIO Stopped and DMA Stopped states, the PCI
host bridge hardware does so in a way that does not introduce failures that might corrupt other
parts of the platform.

11. The PCI host bridge hardware allows firmware access to internal PCI host bridge and I/O fabric
PCI configuration registers when any or all of the partitionable endpoints are in the MMIO
Stopped state.

3.3.2. Partitionable-Endpoint State and Enhanced Error
Handling Test Suite Optional Tests

The Partitionable-Endpoint State and Enhanced Error Handling optional tests need to check all of
the implemented optional functions and verify correct behavior.

Note

There are no Partitionable-Endpoint State and Enhanced Error Handling optional tests

3.4. Error-Injection

The Error-Injection tests ensure that the PCI host bridge hardware and firmware supports the Error-
Injection function. The error-injection hardware is defined primarily to test enhanced error-recovery
software. As implemented in the I/O bridge, this option is used to test the software that implements the recovery that is enabled by the enhanced-error-handling option in that bridge.

### 3.4.1. Error-Injection Test Suite Required Tests

The Error-Injection required tests need to check all of the following required functions and verify correct behavior:

1. The PCI host bridge hardware provides a way to inject the required errors for each partitionable-endpoint primary bus. The errors are injectable independently without affecting the operations on the other buses in the platform.

2. The PCI host bridge hardware provides a way to set up for the injection of the required errors without disturbing operations to other buses outside the partitionable endpoint.

3. The PCI host bridge hardware provides firmware with a way to set up the following information for the error-injection operation by normal processor Load and Store instructions:
   - Address at which to inject the error
   - Address mask to mask off any combination of the least-significant 24 (64 for the ioa-bus-error-64 function) bits of the address
   - Partitionable-endpoint primary bus number that is to receive the error
   - Type of error to be injected

4. The PCI host bridge hardware provides the capability of selecting the errors specified in Supported Errors for PCI Express Error Injectors table of the IODA3 Specification and an indication of when that error is appropriate for the platform configuration.

5. The PCI host bridge hardware provides a way to inject the errors in Supported Errors for PCI Express Error Injectors table of the IODA3 Specification in a non-persistent manner (that is, at most one injection for each invocation of the ibm,errinjct RTAS call).

6. The firmware limits the injection of errors that are inappropriate for the given platform configuration.

### 3.4.2. Error-Injection Test Suite Optional Tests

The Error-Injection optional tests need to check all of the implemented optional functions and verify correct behavior.

**Note**

There are no Error-Injection optional tests

### 3.5. DMA with No Page Migration

The DMA with No Page Migration tests ensure that the PCI host bridge hardware and firmware supports the DMA with No Page Migration function. This includes the translation validation entry (TVE) table in the PCI host bridge chip, the translation control entry (TCE) table in system memory and cached on the PCI host bridge chip, invalidating cached TCEs, I/O Address Validation and TCE Translation.
3.5.1. DMA with No Page Migration Test Suite Required Tests

The DMA with No Page Migration required tests need to check all of the following required functions and verify correct behavior.

1. The PCI host bridge hardware implements the TVE table, as defined by TVE Definition table of the IODA3 Specification with the translation validation table (TVT) being located on the PCI host bridge chip.

2. The PCI host bridge hardware implements the TCE table with entries as defined by TCE Definition table of the IODA3 Specification with the TCE table being in system memory and cached on the PCI host bridge chip.

3. The PCI host bridge hardware provides a TCE Invalidate Register as defined in TCE Invalidate Register Definition table of the IODA3 Specification for invalidating cached TCEs, all TCEs for a particular partitionable endpoint, or the entire cache of TCEs. The hardware stops using the entry when firmware indicates to invalidate, but it can wait until the TCE is used once by a DMA operation. A Store to this register causes the specified operation. Issuing a Load to this register causes the last value stored to be returned.

4. The PCI host bridge hardware implements the DMA flows as shown in DMA Operation High-Level Diagram: No Page Migration figure of the IODA3 Specification, I/O Address Validation and TCE Translation Implementation for 32-Bit DMA Addresses figure of the IODA3 Specification, and I/O Address Validation and TCE Translation Implementation for 64-Bit DMA Addresses figure of the IODA3 Specification.

5. The platform firmware sets up the TVEs appropriately.

6. The platform firmware accesses the TVE table with 8-byte Loads and Stores, naturally aligned.

7. The platform firmware sets up the TCEs appropriately.

8. The platform firmware maintains the coherency between the system memory TCE value and the cached TCE value by using the TCE Invalidate Register to invalidate cached TCEs whenever it changes the value of a TCE.

3.5.2. DMA with No Page Migration Test Suite Optional Tests

The DMA with No Page Migration optional tests need to check all of the implemented optional functions and verify correct behavior.

Note
There are no DMA with No Page Migration optional tests

3.6. DMA with Page Migration

The DMA with Page Migration tests ensure that the PCI host bridge hardware and firmware supports the DMA with Page Migration function. The page-migration facilities in the PCI host bridge hardware
give the firmware the tools necessary to keep DMA operations going while copying a memory page from a source location to the target location.

3.6.1. DMA with Page Migration Test Suite Required Tests

The DMA with Page Migration required tests need to check all of the following required functions and verify correct behavior.

1. The PCI host bridge hardware implements a set of Migration Registers, with the definition in Migration Register Definition table of the IODA3 Specification. The number of registers implemented is implementation dependent, but the minimum number is 7.

2. The PCI host bridge hardware uses the Page Mapping and Control field from the TCE for both source page and target page operations.

3. The PCI host bridge hardware implements the Migration Pointer field of the TCE, as specified in TCE Definition table of the IODA3 Specification. When that field is nonzero, perform the operations described by 4 \[13\] through 9 \[13\] of this requirement.

4. The PCI host bridge hardware accesses the TCE for the operation, as for a normal DMA operation (that is, as per DMA Design Details: No Page Migration section of the IODA3 Specification). Calculate the source page address as usual.

5. The PCI host bridge hardware calculates the migration target page address. If N is the value in the Migration Register Target Page Size field, the address is generated by:

\[
\text{Bits 4:}(63 - N) \text{ of the Migration Register} \| \text{ TCE-translated source page address bits (64 - N):63}
\]

6. If the operation is a DMA Read and the Read Target bit in the Migration Register = 0, the PCI host bridge hardware accesses at the Migration Source page address in the translation page address.

7. If the operation is a DMA Read and the Read Target bit in the Migration Register = 1, the PCI host bridge hardware accesses at the Migration Target page address.

8. If the operation is a DMA Write, the PCI host bridge hardware writes the data to the source page at the TCE-translated page address. After this first write is visible to all other processors and mechanisms, write the data to target page address.

9. The PCI host bridge hardware prevents a PCIe atomic operation that targets a page being migrated from being performed until the migration operation is completed against the page being targeted by the PCIe atomic operation (that is, until the Migration Pointer in the TCE for the page is set to 0). Perform this atomic operation blocking without blocking DMA read and DMA write requests.

10. The platform firmware allocates a Migration Descriptor Register in each PCI host bridge with TCE accessibility to the source page, for use in the migration.

11. The platform firmware builds the Migration Descriptor Register content for the physical page of the memory area to be migrated, and sets the Read Target bit to zero. If there are multiple
page mappings that map the physical page, there only needs to be one Migration Descriptor Register set up for all the page mappings (for example, a 64 KB page migration with multiple 4 KB pages mapped within that 64 KB page). Firmware ensures that the Target Page Size field in the Migration Register is equal to the largest I/O page size being migrated.


13. The platform firmware redirects all TCEs pointing to the memory area to be migrated to the relevant Migration Descriptor Registers. Then, use the TCE Invalidate facility to invalidate any cached versions of the TCE (as per the usual TCE change process).

14. The platform firmware copies data from the source to the destination memory region. For each atomically writable quantum of memory:
   - Read quantum from the migration source page.
   - Write quantum to the corresponding offset in the migration destination page.
   - Reread quantum from the migration source page.
   - Compare the first and second reads (this catches a DMA write race).
   - If not equal, branch back and copy again.
   - Else, loop to the next quantum.

15. The platform firmware sets the Read Target Bit to 1 in each Migration Descriptor Register allocated in 10 [13] of this requirement.

16. The platform firmware adjusts all TCEs changed in 13 [14] of this requirement to directly access their migration destination pages. Set the Migration Pointer in those TCEs to 0. Then, use the TCE Invalidate facility to invalidate any cached versions of the TCE (as per the usual TCE change process).

3.6.2. DMA with Page Migration Test Suite Optional Tests

The DMA with Page Migration optional tests need to check all of the implemented optional functions and verify correct behavior.

**Note**

There are no DMA with Page Migration optional tests.

3.7. DMA with Multilevel TCE Tables

The DMA with Multilevel TCE Tables tests ensure that the PCI host bridge hardware and firmware supports the DMA with Multilevel TCE Tables function. This includes multilevel TCE tables, indirect TCE tables, direct TCE tables, and responding to page faults appropriately.

3.7.1. DMA with Multilevel TCE Tables Test Suite Required Tests

The DMA with Multilevel TCE Tables required tests need to check all of the following required functions and verify correct behavior.
1. The PCI host bridge hardware implements multilevel TCE tables as defined by DMA Design Details: Multilevel TCE Tables section of the IODA3 Specification.

2. The PCI host bridge hardware implements the Number of TCE Table Levels field according to the definition in TVE Definition table of the IODA3 Specification. Use this value to determine when an indirect TCE table is being accessed and when a direct TCE table is being accessed.

3. The PCI host bridge hardware treats a Page Fault setting (0b00) of the TCE Page Mapping and Control field the same for both direct and indirect TCEs. That is, fail the operation and set the enhanced-error-handling Stopped state for the partitionable endpoint.

4. Except for the Page Fault setting (0b00) of the TCE Page Mapping and Control field, the PCI host bridge hardware ignores any read-only or write-only setting for these bits in indirect TCEs. That is, the values of 0b01, 0b10, and 0b11 are to be treated the same for indirect TCEs; they are all valid states for any DMA operation.

5. The PCI host bridge hardware treats the TCE Table Size of the TVE as the size of each level of TCE table being accessed by the TCE table address (TTA) of the TVE or by the Real Page Number (RPN) of an indirect TCE.

6. The PCI host bridge hardware validates that the All-0's field of the PCIe address, shown in PCIe Normal DMA Operation for a Three-Level TCE Table figure of the IODA3 Specification, is all zeros, based on the TCE Table Size, Number of TCE Table Levels, and I/O Page Size fields of the TVE.

7. The platform firmware sets up the Number of TCE Table Levels field appropriately for each TVE.

8. The platform firmware sets up the indirect TCE tables appropriately such that each indirect TCE points to the start of the next level TCE table to be accessed. Set the Page Mapping and Control field for used indirect TCEs to something other than the Page Fault (0b00) setting.

**3.7.2. DMA with Multilevel TCE Tables Test Suite**

**Optional Tests**

The DMA with Multilevel TCE Tables optional tests need to check all of the implemented optional functions and verify correct behavior.

**Note**

There are no DMA with Multilevel TCE Tables optional tests

**3.8. DMA Read Sync Register**

The DMA Read Sync Register tests ensure that the PCI host bridge hardware and firmware supports the DMA Read Sync Register function. The DMA Read Sync Register is provided to assist firmware in determining when all currently outstanding (in progress relative to the PCI host bridge’s state machine) DMA read operations are complete. For example, it can be used during a memory-migration operation or during partitionable-endpoint-reset operations to assure that in-flight DMAs are complete.
3.8.1. DMA Read Sync Register Test Suite Required Tests

The DMA Read Sync Register required tests need to check all of the following required functions and verify correct behavior.

1. The PCI host bridge hardware provides the DMA Read Sync register, as defined in DMA Read Sync Register table of the IODA3 Specification.

3.8.2. DMA Read Sync Register Test Suite Optional Tests

The DMA Read Sync Register optional tests need to check all of the implemented optional functions and verify correct behavior.

Note

There are no DMA Read Sync Register optional tests

3.9. Message-Signalled Interrupt

The Message-Signalled Interrupt tests ensure that the PCI host bridge hardware and firmware supports the Message-Signalled Interrupt function. This includes interrupt vector table (IVT) with entries defined as MSI interrupt vector entry (IVE), IVT BAR and IVT Length, address to access the IVE, partitionable-endpoint number inspection, event trigger forwarding to the Interrupt Virtualization Source Engine, interrupt vector cache (IVC) cache for caching IVEs for MSI operations, and method to invalidate cached IVEs.

3.9.1. Message-Signalled Interrupt Test Suite Required Tests

The Message-Signalled Interrupt required tests need to check all of the following required functions and verify correct behavior.

1. The PCI host bridge hardware decodes MSI operations by detecting that PCIe addresses bits 61:60 are equal to 0b01, for 64-bit I/O DMAs, or that PCIe address bits 31:16 are equal to 0xFFFF (a 64 KB region), for 32-bit I/O DMA addresses.

Note

It is permissible for implementations to decode, under configuration control, additional PCIe address ranges as MSI operations.

2. The PCI host bridge hardware disables 32-bit MSIs by default and provides firmware with a way to enable them.

3. The PCI host bridge hardware implements the IVT with the entries defined as in MSI IVE Definition table of the IODA3 Specification.
4. The PCI host bridge hardware implements IVT BAR and IVT Length registers, writable by firmware, whose contents point to the start and length, respectively, of the IVT in system memory.

5. The PCI host bridge hardware creates the address to access the IVE for the MSI operation by ORing together the following entities:
   - IVT BAR
   - “n” low-order PCIe address bits, where 2^n is the IVT Length, aligned with the low-order IVT BAR bits
   - PCIe data bits 4:0 aligned with PCIe address bits 8:4

6. The PCI host bridge hardware accesses the IVE with the address generated in 5 [17]. Then compare the PE# field from the IVE with the partitionable-endpoint number generated in Requirement R1-3.2.1.2-1 in the IODA3 specification. If they are not equal, put the partitionable endpoint into the MMIO and DMA Stopped states, as defined in Partitionable-Endpoint State and Enhanced Error Handling section of the IODA3 Specification.

7. The PCI host bridge hardware forwards the event trigger to the Interrupt Virtualization Source Engine.

8. The PCI host bridge hardware provides an IVC cache for caching IVEs used for MSI operations.

9. The PCI host bridge hardware provides an IVC Invalidate Register as defined in IVC Invalidate Register Definition table of the IODA3 Specification, for invalidating IVC entries. The hardware stops using the entry when firmware indicates to invalidate. However, it can wait until any invalidated IVE is used once. A Store to this register causes the specified IVE or all IVEs in a cache line to be invalidated. Issuing a Load to this register causes the last value stored to be returned.

10. The platform firmware sets up the IVT BAR and IVT Length Registers in the PCI host bridge appropriately to point to the IVT. The IVT length is set to a power of 2. The IVT BAR alignment is equal to an integer multiple of the size.

11. The platform firmware sets up the IVE for each interrupt source and initializes the corresponding Event State Buffer entries to the Off state.


13. The platform firmware does not allow interrupts to be shared between partitionable endpoints.

14. The platform firmware maintains the coherency between the system memory IVE contents and the cached IVE contents by using the IVC Invalidate Register to invalidate cached IVEs whenever it changes the value of an IVE in system memory.

3.9.2. Message-Signalled Interrupt Test Suite Optional Tests

The Message-Signalled Interrupt optional tests need to check all of the implemented optional functions and verify correct behavior.

Note

There are no Message-Signalled Interrupt optional tests
3.10. PCIe Configuration Space

The PCIe Configuration Space tests ensure that the PCI host bridge hardware and firmware supports the PCIe Configuration Space function. Firmware must be able to access the PCIe configuration space while the other partitionable endpoints remain in the MMIO Stopped state.

3.10.1. PCIe Configuration Space Test Suite Required Tests

The PCIe Configuration Space required tests need to check all of the following required functions and verify correct behavior.

1. The PCI host bridge hardware provides configuration access to an I/O adapter function and to the I/O fabric at all times to the firmware, even though the MMIO Stopped state for a partitionable endpoint might be set. That is, the hardware provides partitionable endpoints for the configuration space that are separate from the normal MMIO memory space partitionable endpoints.

3.10.2. PCIe Configuration Space Test Suite Optional Tests

The PCIe Configuration Space optional tests need to check all of the implemented optional functions and verify correct behavior.

Note

There are no PCIe Configuration Space optional tests

3.11. Partitionable-Endpoint State Table

The Partitionable-Endpoint State Table tests ensure that the PCI host bridge hardware and firmware supports the Partitionable-Endpoint State Table function. This includes partitionable-endpoint state table (PEST) in system memory, PEST BAR, partitionable-endpoint number index into the PEST, and error information recording into the partitionable-endpoint state entry (PESE).

3.11.1. Partitionable-Endpoint State Table Test Suite Required Tests

The Partitionable-Endpoint State Table required tests need to check all of the following required functions and verify correct behavior.

1. The PCI host bridge hardware implements the PEST in system memory, with entries defined by PESE Definition table of the IODA3 Specification.

2. The PCI host bridge hardware implements a BAR, to point to the start of the PEST (PEST BAR), that is loadable by the firmware.

3. The PCI host bridge hardware uses the partitionable-endpoint number, with four trailing zeros concatenated, to index into the PEST.
4. When a partitionable endpoint is placed into the MMIO Stopped state, the PCI host bridge hardware writes the appropriate error information into the PESE for that partitionable-endpoint number.

5. The platform firmware sets up the PEST BAR to point to the start of the PEST in contiguous real system memory, with a size that is a power of 2 and with an address alignment on an integer multiple of the size of the table.

6. The platform firmware clears the contents of a partitionable-endpoint lookup entry (PELE) corresponding to a partitionable-endpoint number before clearing the MMIO Stopped state for that partitionable endpoint.

3.11.2. Partitionable-Endpoint State Table Test Suite Optional Tests

The Partitionable-Endpoint State Table optional tests need to check all of the implemented optional functions and verify correct behavior.

**Note**

There are no Partitionable-Endpoint State Table optional tests.

3.12. Successful Execution of the Required Tests

The template test harness or any other test harness should indicate a Return Code 0 for successful completion with no errors (including no hangs and no machine checks). It should print error messages indicating what failed if there is a non-zero return code. If there is a non-zero return code the design should be debugged and fixed so that the tests can be run again.

3.13. Successful Execution of the Optional Tests

The template test harness or any other test harness should indicate a Return Code 0 for successful completion with no errors (including no hangs and no machine checks). It should print error messages indicating what failed if there is a non-zero return code. If there is a non-zero return code the design should be debugged and fixed so that the tests can be run again.
Appendix A. Glossary

BAR  Base Address Register.
DLPAR Dynamic logical partitioning.
DMA  Direct memory access.
EEH  Enhanced error handling.
IODA3 I/O Design Architecture, version 3.
IVC  Interrupt vector cache.
IVE  Interrupt vector entry.
IVT  Interrupt vector table. A table of IVEs.
MMIO Memory-mapped I/O.
MSI Message signalled interrupt.
MSI-X Message signalled interrupt - extended.
PCI  Peripheral Component Interface.
PCIe PCI Express.
PE# Partitionable-endpoint number.
PELE Partitionable-endpoint lookup entry.
PELE-V Partitionable-endpoint lookup entry (vector). The RTT associates a Bus/Device/Func number of an incoming PCIe transaction to either a partitionable-endpoint number or an index into the PELT-V table when the operation is an error message. The PELE-V contains a vector of bits indicating which partitionable-endpoint numbers are affected by the incoming RID.
PESE Partitionable-endpoint state entry.
PEST Partitionable-endpoint state table.
PHB PCI-based host bridge. An entity that attaches a PCIe bus to the system.
PHB chip The hardware chip where the PCI host bridge is implemented. The PCI host bridge might only be part of the chip functionality; for example, when the PCI host bridge is implemented on the processor chip. In that case, the processor chip becomes the PCI host bridge chip for purposes of this architecture.
RID Requester ID. A name for the combined Bus/Dev/Func fields. The RID is attached to each PCIe transaction. It uniquely identifies the requester of the transaction. Given the uniqueness of this identifier, it is used by IODA3 to separate facilities in the PCI host bridge that are unique to the Func requesting the operation (for example, address translation, interrupt validation, and so on).
RPN Real page number. The bits in the TCE that are used to replace the high-order I/O bus address bits.
RTAS Run-Time Abstraction Services.
RTC RID translation cache. An optional PCI host bridge implementation that allows for better PCI transaction performance when the RTT is in system memory.
RTE RID translation entry. An entry in the RTT.
RTT RID translation table. A 64K-entry table that takes the 16-bit RID from a PCI transaction and maps that to a partitionable-endpoint number (for DMA and MSI operations) or to a PELE-V (when the operation is an error message).
TCE Translation control entry. Used to translate an I/O address page number to a real page number in system memory.
TTA TCE table address. The address of the start of the TCE table. It is contained in the TVE.
TVE Translation validation entry. An entry in a TTV. Used to translate and validate an I/O adapter’s access to a DMA address space.
TVT Translation validation table (in the PCI host bridge). A table containing TVEs.
UR Unsupported request.
Appendix B. OpenPOWER Foundation overview

The OpenPOWER Foundation was founded in 2013 as an open technical membership organization that will enable data centers to rethink their approach to technology. Member companies are enabled to customize POWER CPU processors and system platforms for optimization and innovation for their business needs. These innovations include custom systems for large or warehouse scale data centers, workload acceleration through GPU, FPGA or advanced I/O, platform optimization for SW appliances, or advanced hardware technology exploitation. OpenPOWER members are actively pursing all of these innovations and more and welcome all parties to join in moving the state of the art of OpenPOWER systems design forward.

To learn more about the OpenPOWER Foundation, visit the organization website at openpowerfoundation.org.

B.1. Foundation documentation

Key foundation documents include:

- *Bylaws of OpenPOWER Foundation*
- *OpenPOWER Foundation Intellectual Property Rights (IPR) Policy*
- *OpenPOWER Foundation Membership Agreement*
- *OpenPOWER Anti-Trust Guidelines*

More information about the foundation governance can be found at openpowerfoundation.org/about-us/governance.

B.2. Technical resources

Development resources fall into the following general categories:

- *Technical Steering Committee*
- *Foundation work groups*
- *OpenPOWER Ready documentation, products, and certification criteria*
- *Resource Catalog*

To find all OpenPOWER resources of the following types, select the specified combination of *Resource Type/Main Category/Sub-category* in the Resource Catalog:

<table>
<thead>
<tr>
<th>Specifications</th>
<th>Developer Resources/OpenPOWER Documents/Specifications</th>
</tr>
</thead>
<tbody>
<tr>
<td>Work Group Notes</td>
<td>Developer Resources/OpenPOWER Documents/Work Group Notes</td>
</tr>
</tbody>
</table>
B.3. Contact the foundation

To learn more about the OpenPOWER Foundation, please use the following contact points:

• General information -- <info@openpowerfoundation.org>
• Membership -- <membership@openpowerfoundation.org>
• Technical Work Groups and projects -- <tsc-chair@openpowerfoundation.org>
• Events and other activities -- <admin@openpowerfoundation.org>
• Press/Analysts -- <press@openpowerfoundation.org>

More contact information can be found at openpowerfoundation.org/get-involved/contact-us.