

OpenPOWER ELFv2 Errata

For ELFv2 ABI Version 1.4

Workgroup Notes

Revision 1.0 (March 13, 2018)



www.openpowerfoundation.org

OpenPOWER ELFv2 Errata: For ELFv2 ABI Version 1.4

System Software Work Group <syssw-chair@openpowerfoundation.org>

OpenPower Foundation

Revision 1.0 (March 13, 2018)

Copyright © 2018 OpenPOWER Foundation

Licensed under the GNU Free Documentation License, Version 1.3; with no Invariants Sections, with no Front-Cover Texts, and with no Back-Cover Texts (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

<http://www.gnu.org/licenses/fdl-1.3.txt>

The limited permissions granted above are perpetual and will not be revoked by OpenPOWER or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis AND TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, THE OpenPOWER Foundation AS WELL AS THE AUTHORS AND DEVELOPERS OF THIS STANDARDS FINAL DELIVERABLE OR OTHER DOCUMENT HEREBY DISCLAIM ALL OTHER WARRANTIES AND CONDITIONS, EITHER EXPRESS, IMPLIED OR STATUTORY, INCLUDING BUT NOT LIMITED TO, ANY IMPLIED WARRANTIES, DUTIES OR CONDITIONS OF MERCHANTABILITY, OF FITNESS FOR A PARTICULAR PURPOSE, OF ACCURACY OR COMPLETENESS OF RESPONSES, OF RESULTS, OF WORKMANLIKE EFFORT, OF LACK OF VIRUSES, OF LACK OF NEGLIGENCE OR NON-INFRINGEMENT.

OpenPOWER, the OpenPOWER logo, and openpowerfoundation.org are trademarks or registered trademarks of OpenPOWER Foundation, Inc., registered in many jurisdictions worldwide. Other company, product, and service names may be trademarks or service marks of others.

Abstract

This document provides errata against version 1.4 of the OpenPOWER ELF v2 Application Binary Interface Specification. These errata should be considered part of said specification until such time as a newer version of the full specification is published.

This document is a Non-standard Track, Work Group Note work product owned by the System Software Workgroup and handled in compliance with the requirements outlined in the *OpenPOWER Foundation Work Group (WG) Process* document. It was created using the *Document Development Guide* version 1.1.0. Comments, questions, etc. can be submitted to the public mailing list for the parent specification at <syssw-elfv2abi@mailinglist.openpowerfoundation.org>.

Table of Contents

Preface	iv
1. Conventions	iv
2. Document change history	v
1. Errata	1
1.1. Section 2.1.2.2, Fundamental Types	1
1.2. Section 6.1, Vector Data Types	1
1.3. Section 6.4, Vector Built-In Functions	1
1.4. Section 6.4.1, Big-Endian Vector Layout in Little-Endian Environments	2
A. OpenPOWER Foundation overview	4
A.1. Foundation documentation	4
A.2. Technical resources	4
A.3. Contact the foundation	5

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 \[iv\]](#), references the [Preface](#) chapter on page [iv](#).

2. Document change history

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

The following table describes the most recent changes:

Revision Date	Summary of Changes
March 13, 2018	<ul style="list-style-type: none">Initial document publication

1. Errata

The following statements in version 1.4 of the 64-Bit ELF V2 ABI Specification for the Power Architecture are incorrect, and should be considered corrected as specified.

1.1. Section 2.1.2.2, Fundamental Types

Problem: Paragraph 2 contains a statement that makes use of undefined behavior according to the C standard.

Regardless of the alignment rules for the allocation of data types, pointers to both aligned and unaligned data of each data type shall return the value corresponding to a data type starting at the specified address when accessed with either the pointer dereference operator `*` or the array reference operator `[]`.

Resolution: Paragraph 2 is stricken from the ABI.

1.2. Section 6.1, Vector Data Types

Problem 1: The following sentence likewise makes use of undefined behavior according to the C standard.

While the preferred alignment for vector data types is a multiple of 16 bytes, pointers may point to vector objects at an arbitrary alignment.

As stated in Table 2.12, Vector Types, *all* vector types have quadword alignment; there is no "preferred" alignment.

Resolution 1: The offending sentence is stricken from the ABI.

Problem 2: This section contains another sentence and an example that likewise make use of undefined behavior according to the C standard. The offending sentence reads:

The use of vector built-in functions such as `vec_xl` and `vec_xst` is discouraged except for languages where no dereference operators are available.

The offending example appears directly afterwards.

Resolution 2: The offending sentence is replaced as follows:

The dereference operator `*` may *not* be used to access data that is not aligned at least to a quadword boundary. Built-in functions such as `vec_xl` and `vec_xst` are provided for unaligned data access.

The offending example is stricken from the ABI.

1.3. Section 6.4, Vector Built-In Functions

Problem 1: The Note following Table 6.1 likewise encourages undefined behavior (in this context):

Reminder: The assignment operator = is the preferred way to assign values from one vector data type to another vector data type in accordance with the C and C++ programming languages.

Resolution 1: The Note is stricken from the ABI.

Problem 2: The paragraph following Table 6.2 ends with the following sentence that likewise encourages undefined behavior:

It is recommended that programmers use the assignment operator = or the vector `vec_xl` and `vec_xst` vector built-in functions to access unaligned data streams.

Resolution 2: This sentence is replaced as follows:

It is recommended that programmers use the `vec_xl` and `vec_xst` vector built-in functions to access unaligned data streams.

Problem 3: The first sentence of the second paragraph following Table 6.2 wrongly identifies `vec_xl` and `vec_xst` as optional built-ins:

The set of extended mnemonics in Table 6.3, "Optional Built-In Memory Access Functions" [127] may be provided by some compilers and are not required by the Power SIMD programming interfaces.

The title of Table 6.3 likewise calls these optional.

Resolution 3: The title of Table 6.3 is changed to "VSX Memory Access Built-In Functions", and this sentence is replaced as follows:

The built-in functions in Table 6.3, "VSX Memory Access Built-In Functions" [127] provide unaligned access to data in memory that is to be copied to or from a variable having vector data type.

Problem 4: The second paragraph following Table 6.2 contains one sentence that likewise encourages undefined behavior, and another sentence that is simply incorrect:

In particular, the assignment operator = will have the same effect of copying values between vector data types and provides a preferable method to assign values while giving the compiler more freedom to optimize data allocation. The only use for these functions is to support some coding patterns enabling big-endian vector layout code sequences in both big-endian and little-endian environments.

The second sentence applies to functions in table 6.4, not 6.3.

Resolution 4: The offending sentences are stricken from the ABI.

1.4. Section 6.4.1, Big-Endian Vector Layout in Little-Endian Environments

Problem: Table 6.8, and the sentence preceding it, wrongly identify the `vec_xl` and `vec_xst` built-in functions as optional.

See Table 6.8, "Optional Built-In Memory Access Functions (BE Layout in LE Mode)" [129].

Resolution: Table 6.8 is renamed "VSX Built-In Memory Access Functions (BE Layout in LE Mode)" and referenced thus in the preceding sentence.

Appendix A. 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 pursuing 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.

A.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.

A.2. Technical resources

Development resources fall into the following general categories:

- [Foundation work groups](#)
- [Remote development environments \(VMs\)](#)
- [Development systems](#)
- [Technical specifications](#)
- [Software](#)
- [Developer tools](#)

The complete list of technical resources are maintained on the foundation [Technical Resources](#) web page.

A.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.