



# Obviating Memory-Management Instructions after Marking PTEs Valid (Svvptc)

Version v1.0, 2024-06-29: Ratified

# Table of Contents

Preamble.....	1
Copyright and license information.....	1
Contributors.....	1
1. Obviating Memory-Management Instructions after Marking PTEs Valid (Svptc) .....	2

# Preamble



*This document is in the [Ratified state](#)*

No changes are allowed. Any desired or needed changes can be the subject of a follow-on new extension. Ratified extensions are never revised.

## Copyright and license information

This specification is licensed under the Creative Commons Attribution 4.0 International License (CC-BY 4.0). The full license text is available at [creativecommons.org/licenses/by/4.0/](https://creativecommons.org/licenses/by/4.0/).

Copyright 2024 by RISC-V International.

## Contributors

This RISC-V specification has been contributed to directly or indirectly by: Alexandre Ghiti, Andrea Parri, Andrew Waterman, David Weaver, Greg Favor, Guy Lemieux, John Henry Deppe, Ken Dockser, Krste Asanovic, Ved Shanbhogue

# Chapter 1. Obviating Memory-Management Instructions after Marking PTEs Valid (Svvptc)

When the Svvpctc extension is implemented, explicit stores by a hart that update the Valid bit of leaf and/or non-leaf PTEs from 0 to 1 and are visible to a hart will eventually become visible within a bounded timeframe to subsequent implicit accesses by that hart to such PTEs.



Svvptc relieves an operating system from executing certain memory-management instructions, such as `SFENCE.VMA` or `SINVAL.VMA`, which would normally be used to synchronize the hart's address-translation caches when a memory-resident PTE is changed from Invalid to Valid. Synchronizing the hart's address-translation caches with other forms of updates to a memory-resident PTE, including when a PTE is changed from Valid to Invalid, requires the use of suitable memory-management instructions. Svvpctc guarantees that a change to a PTE from Invalid to Valid is made visible within a bounded time, thereby making the execution of these memory-management instructions redundant. The performance benefit of eliding these instructions outweighs the cost of an occasional gratuitous additional page fault that may occur.

Depending on the microarchitecture, some possible ways to facilitate implementation of Svvpctc include: not having any address-translation caches, not storing Invalid PTEs in the address-translation caches, automatically evicting Invalid PTEs using a bounded timer, or making address-translation caches coherent with store instructions that modify PTEs.