AMD64 RMP Dirty

Publication # **69203** Revision: **1.00**

Issue Date: **February 2026**

Future AMD APM releases that describe this feature supersede the information in this document

The information presented in this document is for informational purposes only and may contain technical inaccuracies, omissions, and typographical errors. The information contained herein is subject to change and may be rendered inaccurate for many reasons, including but not limited to product and roadmap changes, component and motherboard version changes, new model and/or product releases, product differences between differing manufacturers, software changes, BIOS flashes, firmware upgrades, or the like. Any computer system has risks of security vulnerabilities that cannot be completely prevented or mitigated. AMD assumes no obligation to update or otherwise correct or revise this information. However, AMD reserves the right to revise this information and to make changes from time to time to the content hereof without obligation of AMD to notify any person of such revisions or changes. THIS INFORMATION IS PROVIDED "AS IS." AMD MAKES NO REPRESENTATIONS OR WARRANTIES WITH RESPECT TO THE CONTENTS HEREOF AND ASSUMES NO RESPONSIBILITY FOR ANY INACCURACIES, ERRORS, OR OMISSIONS THAT MAY APPEAR IN THIS INFORMATION. AMD SPECIFICALLY DISCLAIMS ANY IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY, OR FITNESS FOR ANY PARTICULAR PURPOSE. IN NO EVENT WILL AMD BE LIABLE TO ANY PERSON FOR ANY RELIANCE, DIRECT, INDIRECT, SPECIAL, OR OTHER CONSEQUENTIAL DAMAGES ARISING FROM THE USE OF ANY INFORMATION CONTAINED HEREIN, EVEN IF AMD IS EXPRESSLY ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

AMD, the AMD Arrow logo, AMD EPYC, and combinations thereof are trademarks of Advanced Micro Devices, Inc.

Other product names used in this publication are for identification purposes only and may be trademarks of their respective companies. PCIe® is a registered trademark of PCI-SIG Corporation. Linux® is the registered trademark of Linus Torvalds in the U.S. and other countries.

Revision History

Date	Revision	Change Description
February 2026	1.00	Initial release

RMP Dirty

Introduction

This document describes the RMP Dirty feature and the RMPCHKD instruction.

Presence

CPUID Fn80000025_EDX[RmpDirty] (bit 2) indicates RMP Dirty and RMPCHKD instruction support.

RMP Dirty

The RMP Dirty feature allows software to track whether an SEV-SNP guest private page has been written since the last time it was marked not-dirty. Each guest private page RMP entry contains a Not-Dirty bit which is reset to 0. At VMPL0, the RMPADJUST instruction writes RDX[17] to the Not-Dirty bit in the page's RMP entry and the RMPQUERY instruction returns the value of the RMP Not-Dirty bit in RDX[17].

The Not-Dirty bit is cleared by the processor the first time there is a write to the page. Additionally, the Not-Dirty bit is cleared by the RMPADJUST instruction if not running at VMPL0 and the PVALIDATE instruction.

RMPCHKD Instruction

The RMPCHKD instruction checks one or more guest private 4KB pages to determine if the page is marked dirty in the RMP. A page is considered dirty if the Not-Dirty bit is 0. The guest physical address of the first 4KB page to check is specified in RAX and the number of 4KB pages to check is specified in RCX. The RMPCHKD instruction completes when either a dirty page is found, or all pages have been checked. RCX is decremented by 1 and RAX is incremented by 0x1000 after each iteration that finds a page marked not-dirty.

When the instruction completes, RFLAGS.ZF indicates if a dirty page was found. If RFLAGS.ZF is 1, no dirty pages were found. If RFLAGS.ZF is 0, a dirty page was found. In this case, RAX contains the guest physical address of the dirty page and RFLAGS.CF indicates the size of that page in the RMP entry. If RFLAGS.CF is 1, it indicates a 2MB page, while 0 indicates a 4KB page. If no dirty page is found, RFLAGS.CF is cleared to 0.

Guest pages are assumed to be private during nested page table translations. The RMPCHKD instruction will take a #VMEXIT(NPF) if a nested translation error occurs or the translated address is outside the range of memory covered by the RMP.

The RMPCHKD instruction can be suspended by an exception or interrupt. When this happens, RAX will hold the value of the next guest physical page, RCX will hold the remaining number of 4KB pages to check and RIP will continue to point to the RMPCHKD instruction, allowing the instruction to be resumed after return from the exception or interrupt handler.

This is a privileged instruction. Attempted execution at a privilege level other than CPL0 will result in a #GP(0) exception. In addition, this instruction is only valid in 64-bit mode in an SNP-active (SEV-SNP) guest; in all other modes a #UD exception will be generated.

Mnemonic	**Opcode**	**Description**
RMPCHKD	F3 0F 01 FC	Checks RMP Dirty status of guest pages

rFlags Affected

ID	VIP	VIF	AC	VM	RF	NT	IOPL	OF	DF	IF	TF	SF	ZF	AF	PF	CF
								U				U	M	U	U	M
21	20	19	18	17	16	14	13:12	11	10	9	8	7	6	4	2	0

Exceptions

Exception	Real	Virtual 8086	Protected	Cause of Exception
Invalid opcode, #UD	X	X	X	Instruction not supported as indicated by CPUID Fn80000025_EDX[RmpDirty] = 0
	X	X	X	This instruction is only recognized in 64-bit mode
			X	Guest is not SNP-Active
General Protection, #GP			X	CPL was not 0
			X	Current VMPL was not zero
VMM Communication, #VC			X	RMP.VALIDATED was not set to 1 Error code GPA_NOT_VALIDATED (0x408)