AMD64 Bit Matrix Multiply and Bit Reversal Instructions

Publication # **69192**	Revision: **1.00**
Issue Date: **January 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.

© 2026 Advanced Micro Devices, Inc. All rights reserved.

Revision History

Date	Revision	Change Description
January 2026	1.00	Initial release

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

Bit Matrix Multiply and Bit Reversal Instructions

The AMD Bit Matrix Multiply and Bit Reversal extension consists of three instructions which are described in detail in the following sections:

- **VBMACOR16X16X16 –** Bit Matrix Multiply and Accumulate with OR Reduction
- **VBMACXOR16X16X16 –** Bit Matrix Multiply and Accumulate with XOR Reduction
- **VBITREV –** Reverse Bits in each Byte

CPUID Function 80000021_EAX[AVX512_BMM] (bit 23) = 1 indicates support for AMD Bit Matrix Multiply and Bit Reversal instructions.

VBMACOR16X16X16 – Bit Matrix Multiply and Accumulate with OR Reduction

Multiply two 16x16 bit matrices using OR reduction and OR the product into a third 16x16 bit matrix, which is also the destination.

For the 256-bit YMM form, the source registers/memory each contain a single 16X16 (256) bit matrix in bits [255:0]. For the 512-bit ZMM form, the registers each contain two 16X16 (256) bit matrices in bits [255:0] and [511:255].

The exact functionality of a bit matrix multiply and accumulate is as follows:

```
for i in 0 to 15
  for j in 0 to 15
    reduction_bit = src1[16*i+j]
    for k in 0 to 15
      reduction_bit |= src2[16*i+k] & src3[16*k+j]
    end for k
    dest[16*i+j] = reduction_bit
  end for j
end for i
```

There are only extended forms of this instruction:

VBMACOR16X16X16

The extended form of the instruction has 256-bit and 512-bit encodings:

YMM Encoding: The first source operand, which is also the destination, is a YMM register. The second source operand is a YMM register. The third source operand is a YMM register or a 256-bit memory location. Bits above bit 255 of the ZMM register corresponding to the destination are cleared.

ZMM Encoding: The first source operand, which is also the destination, is a ZMM register. The second source operand is a ZMM register. The third source operand is a ZMM register or a 512-bit memory location.

The EVEX encoded versions of this instruction do not allow a k-mask to be specified and do not support memory fault suppression.

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

Instruction Form	Instruction Subset
VBMACOR16X16X16	AVX512_BMM (CPUID Function 8000_0021 EAX [23])

Mnemonic	Encoding				
	EVEX	RXBR'.Map	W.vvvv.pp	z.L'L.b.V'.aaa	Opcode
VBMACOR16X16X16 ymm1, ymm2, ymm3/mem256	62	RXBR'.6	0.vvvv.00	0.01.0.V'.000	80 /r
VBMACOR16X16X16 zmm1, zmm2, zmm3/mem512	62	RXBR'.6	0.vvvv.00	0.10.0.V'.000	80 /r

Related Instructions

VBMACXOR16X16X16 VBITREV

rFLAGS Affected

None

MXCSR Flags Affected

None

Exceptions

Exceptions for AVX512 instructions are enumerated in "*AMD64 Architecture Programmer's Manual*", Volume 4 (Revision 3.26 or later), Section 3.2 "AVX512 Exceptions".

VBMACXOR16X16X16 – Bit Matrix Multiply and Accumulate with XOR Reduction

Multiply two 16x16 bit matrices using XOR reduction and XOR the product into a third 16x16 bit matrix, which is also the destination.

For the 256-bit YMM form, the source registers/memory each contain a single 16X16 (256) bit matrix in bits [255:0]. For the 512-bit ZMM form, the registers each contain two 16X16 (256) bit matrices in bits [255:0] and [511:255].

The exact functionality of a bit matrix multiply and accumulate is as follows:

```
for i in 0 to 15
  for j in 0 to 15
    reduction_bit = src1[16*i+j]
    for k in 0 to 15
      reduction_bit ^= src2[16*i+k] & src3[16*k+j]
    end for k
    dest[16*i+j] = reduction_bit
  end for j
end for i
```


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

There are only extended forms of this instruction:

VBMACOR16X16X16

The extended form of the instruction has 256-bit and 512-bit encodings:

YMM Encoding: The first source operand, which is also the destination, is a YMM register. The second source operand, is a YMM register. The third source operand, is a YMM register or a 256-bit memory location. Bits above bit 255 of the ZMM register corresponding to the destination are cleared.

ZMM Encoding: The first source operand, which is also the destination, is a ZMM register. The second source operand, is a ZMM register. The third source operand, is a ZMM register or a 512-bit memory location.

The EVEX encoded versions of this instruction do not allow a k-mask to be specified and do not support memory fault suppression.

Instruction Form	Instruction Subset
VBMACOR16X16X16	AVX512_BMM (CPUID Function 8000_0021 EAX [23])

Mnemonic	Encoding				
	EVEX	RXBR'.Map	W.vvvv.pp	z.L'L.b.V'.aaa	Opcode
VBMACOR16X16X16 ymm1, ymm2, ymm3/mem256	62	RXBR'.6	1.vvvv.00	0.01.0.V'.000	80 /r
VBMACOR16X16X16 zmm1, zmm2, zmm3/mem512	62	RXBR'.6	1.vvvv.00	0.10.0.V'.000	80 /r

Related Instructions

VBMACOR16X16X16 VBITREV

rFLAGS Affected

None

MXCSR Flags Affected

None

Exceptions

Exceptions for AVX512 instructions are enumerated in "*AMD64 Architecture Programmer's Manual*", Volume 4 (Revision 3.26 or later), Section 3.2 "AVX512 Exceptions".

VBITREV – Reverse Bits in each Byte

For each byte in the source, reverse the bits in that byte to generate the corresponding destination byte.

There are only extended forms of this instruction:

VBITREV

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

The extended form of the instruction has 128-bit, 256-bit and 512-bit encodings:

XMM Encoding: The source operand is an XMM register or a 128-bit memory location. The destination is an XMM register. Bits above bit 127 of the YMM/ZMM register corresponding to the destination are cleared.

YMM Encoding: The source operand is a YMM register or a 256-bit memory location.The destination is a YMM register. Bits above bit 255 of the ZMM register corresponding to the destination are cleared.

ZMM Encoding: The source operand is a ZMM register or a 512-bit memory location. The destination is a ZMM register.

The EVEX encoded versions of this instruction write to the destination based on the supplied k mask per 8-bit element and support memory fault suppression.

Instruction Form	Instruction Subset
VBITREV	AVX512_BMM (CPUID Function 8000_0021 EAX [23])

Mnemonic	Encoding				
	EVEX	RXBR'.Map	W.vvvv.pp	z.L'L.b.V'.aaa	Opcode
VBITREV xmm1 {k}{z}, xmm2/mem128	62	RXBR'.6	0.1111.00	z.00.0.1.aaa	81 /r
VBITREV ymm1 {k}{z}, ymm2/mem256	62	RXBR'.6	0.1111.00	z.01.0.1.aaa	81 /r
VBITREV zmm1 {k}{z}, zmm2/mem512	62	RXBR'.6	0.1111.00	z.10.0.1.aaa	81 /r

Related Instructions

VBMACOR16X16X16 VBMACXOR16X16X16 VPOPCNTB

rFLAGS Affected

None

MXCSR Flags Affected

None

Exceptions

Exceptions for AVX512 instructions are enumerated in "*AMD64 Architecture Programmer's Manual*", Volume 4 (Revision 3.26 or later), Section 3.2 "AVX512 Exceptions".

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