

HADDPS—Packed Single-FP Horizontal Add

Opcode/ Instruction	Op/ En	64/32-bit Mode	CPUID Feature Flag	Description
F2 0F 7C /r HADDPS <i>xmm1</i> , <i>xmm2/m128</i>	RM	V/V	SSE3	Horizontal add packed single-precision floating-point values from <i>xmm2/m128</i> to <i>xmm1</i> .
VEX.128.F2.0F.WIG 7C /r VHADDPS <i>xmm1</i> , <i>xmm2</i> , <i>xmm3/m128</i>	RVM	V/V	AVX	Horizontal add packed single-precision floating-point values from <i>xmm2</i> and <i>xmm3/mem</i> .
VEX.256.F2.0F.WIG 7C /r VHADDPS <i>ymm1</i> , <i>ymm2</i> , <i>ymm3/m256</i>	RVM	V/V	AVX	Horizontal add packed single-precision floating-point values from <i>ymm2</i> and <i>ymm3/mem</i> .

Instruction Operand Encoding

Op/En	Operand 1	Operand 2	Operand 3	Operand 4
RM	ModRM:reg (r, w)	ModRM:r/m (r)	NA	NA
RVM	ModRM:reg (w)	VEX.vvvv (r)	ModRM:r/m (r)	NA

Description

Adds the single-precision floating-point values in the first and second dwords of the destination operand and stores the result in the first dword of the destination operand.

Adds single-precision floating-point values in the third and fourth dword of the destination operand and stores the result in the second dword of the destination operand.

Adds single-precision floating-point values in the first and second dword of the source operand and stores the result in the third dword of the destination operand.

Adds single-precision floating-point values in the third and fourth dword of the source operand and stores the result in the fourth dword of the destination operand.

In 64-bit mode, use of the REX.R prefix permits this instruction to access additional registers (XMM8-XMM15).

See Figure 3-18 for HADDPS; see Figure 3-19 for VHADDPS.

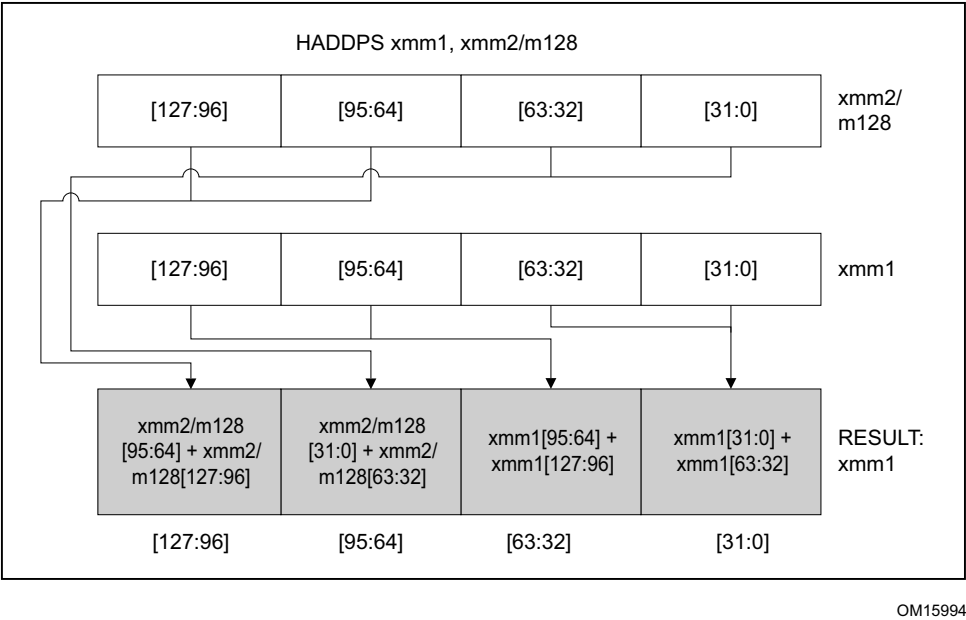


Figure 3-18. HADDPS—Packed Single-FP Horizontal Add

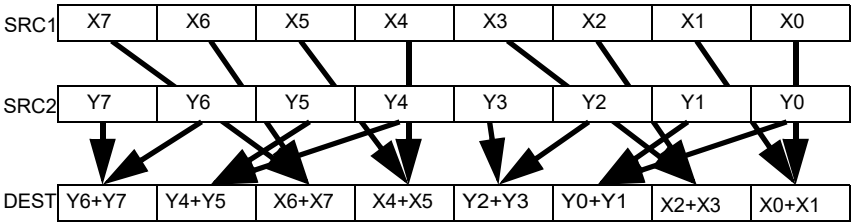


Figure 3-19. VHADDPS operation

128-bit Legacy SSE version: The second source can be an XMM register or a 128-bit memory location. The destination is not distinct from the first source XMM register and the upper bits (`MAXVL-1:128`) of the corresponding YMM register destination are unmodified.

VEX.128 encoded version: the first source operand is an XMM register or 128-bit memory location. The destination operand is an XMM register. The upper bits (`MAXVL-1:128`) of the corresponding YMM register destination are zeroed.

VEX.256 encoded version: The first source operand is a YMM register. The second source operand can be a YMM register or a 256-bit memory location. The destination operand is a YMM register.

Operation

HADDPS (128-bit Legacy SSE version)

$DEST[31:0] \leftarrow SRC1[63:32] + SRC1[31:0]$
 $DEST[63:32] \leftarrow SRC1[127:96] + SRC1[95:64]$
 $DEST[95:64] \leftarrow SRC2[63:32] + SRC2[31:0]$
 $DEST[127:96] \leftarrow SRC2[127:96] + SRC2[95:64]$
 $DEST[MAXVL-1:128]$ (Unmodified)

VHADDPS (VEX.128 encoded version)

$DEST[31:0] \leftarrow SRC1[63:32] + SRC1[31:0]$
 $DEST[63:32] \leftarrow SRC1[127:96] + SRC1[95:64]$
 $DEST[95:64] \leftarrow SRC2[63:32] + SRC2[31:0]$
 $DEST[127:96] \leftarrow SRC2[127:96] + SRC2[95:64]$
 $DEST[MAXVL-1:128] \leftarrow 0$

VHADDPS (VEX.256 encoded version)

$DEST[31:0] \leftarrow SRC1[63:32] + SRC1[31:0]$
 $DEST[63:32] \leftarrow SRC1[127:96] + SRC1[95:64]$
 $DEST[95:64] \leftarrow SRC2[63:32] + SRC2[31:0]$
 $DEST[127:96] \leftarrow SRC2[127:96] + SRC2[95:64]$
 $DEST[159:128] \leftarrow SRC1[191:160] + SRC1[159:128]$
 $DEST[191:160] \leftarrow SRC1[255:224] + SRC1[223:192]$
 $DEST[223:192] \leftarrow SRC2[191:160] + SRC2[159:128]$
 $DEST[255:224] \leftarrow SRC2[255:224] + SRC2[223:192]$

Intel C/C++ Compiler Intrinsic Equivalent

HADDPS: `__m128 _mm_hadd_ps (__m128 a, __m128 b);`

VHADDPS: `__m256 _mm256_hadd_ps (__m256 a, __m256 b);`

Exceptions

When the source operand is a memory operand, the operand must be aligned on a 16-byte boundary or a general-protection exception (#GP) will be generated.

Numeric Exceptions

Overflow, Underflow, Invalid, Precision, Denormal

Other Exceptions

See Exceptions Type 2.