

## General Support Intrinsics

Intrinsic	Operation	Corresponding instruction
<a href="#">mm_empty</a>	Empty register	<code>MOVQ \$0, %mm0</code>
<a href="#">mm_cvtsi32_si64</a>	Convert 32-bit integer to 64-bit	<code>MOVQ %eax, %mm0</code>
<a href="#">mm_cvtsi64_si32</a>	Convert 64-bit integer to 32-bit	<code>MOVQ %mm0, %eax</code>
<a href="#">mm_packs_pi16</a>	Pack 16-bit integers	<code>PACKSSWB</code>
<a href="#">mm_packs_pi32</a>	Pack 32-bit integers	<code>PACKSSDW</code>
<a href="#">mm_packs_pu16</a>	Pack 16-bit unsigned integers	<code>PACKUSWB</code>
<a href="#">mm_unpackhi_pi8</a>	Interleave high bytes	<code>UNPCKHBW</code>
<a href="#">mm_unpackhi_pi16</a>	Interleave high 16-bit words	<code>UNPCKHWD</code>
<a href="#">mm_unpackhi_pi32</a>	Interleave high 32-bit doublewords	<code>UNPCKHDQ</code>
<a href="#">mm_unpacklo_pi8</a>	Interleave low bytes	<code>UNPCKLBW</code>
<a href="#">mm_unpacklo_pi16</a>	Interleave low 16-bit words	<code>UNPCKLWD</code>
<a href="#">mm_unpacklo_pi32</a>	Interleave low 32-bit doublewords	<code>UNPCKLDQ</code>

## Packed Arithmetic Intrinsics

Intrinsic	Operation	Corresponding instruction	Result value bits
<a href="#">mm_add_pi8</a>	Add 8-bit packed integers	<code>PADDQB</code>	8
<a href="#">mm_add_pi16</a>	Add 16-bit packed integers	<code>PADDQW</code>	16
<a href="#">mm_add_pi32</a>	Add 32-bit packed integers	<code>PADDQD</code>	32
<a href="#">mm_adds_pi8</a>	Add 8-bit signed integers	<code>PADDSB</code>	8
<a href="#">mm_adds_pi16</a>	Add 16-bit signed integers	<code>PADDSD</code>	16
<a href="#">mm_adds_pu8</a>	Add 8-bit unsigned integers	<code>PADDUSB</code>	8
<a href="#">mm_adds_pu16</a>	Add 16-bit unsigned integers	<code>PADDUSW</code>	16
<a href="#">mm_sub_pi8</a>	Subtract 8-bit packed integers	<code>PSUBQB</code>	8
<a href="#">mm_sub_pi16</a>	Subtract 16-bit packed integers	<code>PSUBQW</code>	16
<a href="#">mm_sub_pi32</a>	Subtract 32-bit packed integers	<code>PSUBQD</code>	32
<a href="#">mm_subs_pi8</a>	Subtract 8-bit signed integers	<code>PSUBSB</code>	8
<a href="#">mm_subs_pi16</a>	Subtract 16-bit signed integers	<code>PSUBSD</code>	16
<a href="#">mm_subs_pu8</a>	Subtract 8-bit unsigned integers	<code>PSUBUSB</code>	8
<a href="#">mm_subs_pu16</a>	Subtract 16-bit unsigned integers	<code>PSUBUSW</code>	16
<a href="#">mm_madd_pi16</a>	Multiply-add 16-bit integers	<code>PMAQD</code>	32
<a href="#">mm_mulhi_pi16</a>	Multiply high 16-bit integers	<code>MULHW</code>	16
<a href="#">mm_mullo_pi16</a>	Multiply low 16-bit integers	<code>MULLW</code>	16

## Logical Intrinsics

Intrinsic	Operation	Corresponding instruction
<a href="#">mm_and_si64</a>	Bitwise AND	<code>PAND</code>
<a href="#">mm_andnot_si64</a>	Logical AND NOT	<code>PANDN</code>

## MMX Intrinsics - Stefano Tommesani

Written by Stefano Tommesani

Thursday, 27 May 2010 14:08 - Last Updated Monday, 27 May 2013 15:09

---

[mm\\_or\\_si64](#) Bitwise OR  
[mm\\_xor\\_si64](#) Bitwise Exclusive OR

## Compare Intrinsics

Intrinsic Name    Size of data    Instruction  
[mm\\_cmpeq\\_pi8](#)    Equals 8    PCMPEQB  
[mm\\_cmpeq\\_pi16](#)    Equals 16    PCMPEQW  
[mm\\_cmpeq\\_pi32](#)    Equals 32    PCMPEQD  
[mm\\_cmpgt\\_pi8](#)    Greater than    PCMPGTB  
[mm\\_cmpgt\\_pi16](#)    Greater than    PCMPGTW  
[mm\\_cmpgt\\_pi32](#)    Greater than    PCMPGTD

## Set Intrinsics

Intrinsic Name    Size of data    Instruction  
[mm\\_setzero\\_si64](#)    Sets to 0    No  
[mm\\_set\\_pi32](#)    Sets integer to No  
[mm\\_set\\_pi16](#)    Sets integer to No  
[mm\\_set\\_pi8](#)    Sets integer to No  
[mm\\_set1\\_pi32](#)    Sets integer to Yes  
[mm\\_set1\\_pi16](#)    Sets integer to Yes  
[mm\\_set1\\_pi8](#)    Sets integer to Yes  
[mm\\_setr\\_pi32](#)    Sets integer to Yes  
[mm\\_setr\\_pi16](#)    Sets integer to Yes  
[mm\\_setr\\_pi8](#)    Sets integer to Yes

## **MMX Intrinsic - Stefano Tommesani**

Written by Stefano Tommesani

Thursday, 27 May 2010 14:08 - Last Updated Monday, 27 May 2013 15:09

---