

ДОДАТОК А. SSE функції для роботи з даними із плаваючою крапкою

Таблиця А.1 – Арифметичні функції для даних зі звичайною точністю

Ім'я функції	Параметри	Результат	Призначення
<code>_mm_add_ss</code> <code>_mm_sub_ss</code> <code>_mm_mul_ss</code> <code>_mm_div_ss</code>	<code>__m128 a, __m128 b</code>	<code>__m128</code>	$r[0] = a[0] \text{ Q } b[0]$ $r[i] = a[i], i = 1..3$ $Q = \{+, -, *, /\}$
<code>_mm_add_ps</code> <code>_mm_sub_ps</code> <code>_mm_mul_ps</code> <code>_mm_div_ps</code>	<code>__m128 a, __m128 b</code>	<code>__m128</code>	$r[i] = a[i] \text{ Q } b[i]$ $i = 0..3$ $Q = \{+, -, *, /\}$
<code>_mm_hadd_ps</code> (SSE3)	<code>__m128 a, __m128 b</code>	<code>__m128</code>	$r[0] = a[0] + a[1];$ $r[1] = a[2] + a[3];$ $r[2] = b[0] + b[1];$ $r[3] = b[2] + b[3];$
<code>_mm_hsub_ps</code> (SSE3)	<code>__m128 a, __m128 b</code>	<code>__m128</code>	$r[0] = a[0] - a[1];$ $r[1] = a[2] - a[3];$ $r[2] = b[0] - b[1];$ $r[3] = b[2] - b[3];$
<code>_mm_addsub_ps</code> (sse3)	<code>__m128 a,</code> <code>__m128 b,</code>	<code>__m128</code>	$r[0] = a[0] - b[0],$ $r[1] = a[1] + b[1],$ $r[2] = a[2] - b[2],$ $r[3] = a[3] + b[3],$
<code>_mm_sqrt_ss</code>	<code>__m128 a</code>	<code>__m128</code>	$r[0] = \sqrt{a[0]}$ $r[i] = a[i], i = 1..3$
<code>_mm_sqrt_ps</code>	<code>__m128 a</code>	<code>__m128</code>	$r[i] = \sqrt{a[i]}, i = 0..3$
<code>_mm_rcp_ss</code>	<code>__m128 a</code>	<code>__m128</code>	$r[0] = 1/a[0]$ $r[i] = a[i], i = 1..3$
<code>_mm_rcp_ps</code>	<code>__m128 a</code>	<code>__m128</code>	$r[i] = 1/a[i]$ $r[i] = a[i], i = 0..3$
<code>_mm_rsqrt_ss</code>	<code>__m128 a</code>	<code>__m128</code>	$r[0] = 1/\sqrt{a[0]}$ $r[i] = a[i], i = 1..3$
<code>_mm_rsqrt_ps</code>	<code>__m128 a</code>	<code>__m128</code>	$r[i] = 1/\sqrt{a[i]}$ $i = 0..3$
<code>_mm_min_ss</code>	<code>__m128 a, __m128 b</code>	<code>__m128</code>	$r[0] = \min(a[0], b[0])$ $r[i] = a[i], i = 1..3$
<code>_mm_min_ps</code>	<code>__m128 a, __m128 b</code>	<code>__m128</code>	$r[i] = \min(a[i], b[i])$ $i = 0..3$
<code>_mm_max_ss</code>	<code>__m128 a, __m128 b</code>	<code>__m128</code>	$r[0] = \max(a[0], b[0])$ $r[i] = a[i], i = 1..3$
<code>_mm_max_ps</code>	<code>__m128 a, __m128 b</code>	<code>__m128</code>	$r[i] = \max(a[i], b[i]), i = 0..3$

Таблиця А.2 – Арифметичні функції для даних з подвійною точністю

Ім'я функції	Параметри	Результат	Призначення
<code>_mm_add_sd</code> <code>_mm_sub_sd</code> <code>_mm_mul_sd</code> <code>_mm_div_sd</code>	<code>__m128d a, __m128d b</code>	<code>__m128d</code>	$R[0] = a[0] \text{ Q } b[0]$ $r[1] = a[1]$ $Q = \{+, -, *, /\}$
<code>_mm_add_pd</code> <code>_mm_sub_pd</code> <code>_mm_mul_pd</code> <code>_mm_div_pd</code>	<code>__m128d a, __m128d b</code>	<code>__m128d</code>	$R[i] = a[i] \text{ Q } b[i]$ $i = 0..1$ $Q = \{+, -, *, /\}$
<code>_mm_hadd_pd</code>	<code>__m128d a, __m128d b</code>	<code>__m128d</code>	$R[0] = a[0] + a[1];$ $R[1] = b[0] + b[1];$
<code>_mm_hsub_pd</code>	<code>__m128d a, __m128d b</code>	<code>__m128d</code>	$R[0] = a[0] - a[1];$ $R[1] = b[0] - b[1];$
<code>_mm_addsub_pd</code> (sse3)	<code>__m128d a,</code> <code>__m128d b,</code>	<code>__m128d</code>	$R[0] = a[0] - b[0],$ $R[1] = a[1] + b[1],$
<code>_mm_sqrt_sd</code>	<code>__m128d a</code>	<code>__m128d</code>	$r[0] = \sqrt{a[0]}$ $R[1] = a[1]$
<code>_mm_sqrt_pd</code>	<code>__m128d a</code>	<code>__m128d</code>	$r[i] = \sqrt{a[i]}$ $i = 0..1$
<code>_mm_rcp_sd</code>	<code>__m128d a</code>	<code>__m128d</code>	$r[0] = 1/a[0]$ $R[1] = a[1]$
<code>_mm_rcp_pd</code>	<code>__m128d a</code>	<code>__m128d</code>	$r[i] = 1/a[i]$ $i = 0..1$
<code>_mm_rsqrt_sd</code>	<code>__m128d a</code>	<code>__m128d</code>	$r[0] = 1/\sqrt{a[0]}$ $R[1] = a[1]$
<code>_mm_rsqrt_pd</code>	<code>__m128d a</code>	<code>__m128d</code>	$r[i] = 1/\sqrt{a[i]}$ $i = 0..1$
<code>_mm_min_sd</code> <code>_mm_max_sd</code>	<code>__m128d a, __m128d b</code>	<code>__m128d</code>	$r[0] = \ominus(a[0], b[0])$ $R[1] = a[1] (\ominus - \min, \max)$
<code>_mm_min_pd</code> <code>_mm_max_pd</code>	<code>__m128d a, __m128d b</code>	<code>__m128d</code>	$r[i] = \ominus(a[i], b[i])$ $i = 0..1 (\ominus - \min, \max)$

Таблиця А.3 – Функції побітової обробки

Ім'я функції	Параметри	Результат	Призначення
<code>_mm_and_ps</code> <code>_mm_and_pd</code>	<code>__m128 a, __m128 b</code>	<code>__m128</code>	$r = a \& b$
<code>_mm_or_ps</code> <code>_mm_or_pd</code>	<code>__m128 a, __m128 b</code>	<code>__m128</code>	$r = a b$
<code>_mm_xor_ps</code> <code>_mm_xor_pd</code>	<code>__m128 a, __m128 b</code>	<code>__m128</code>	$r = a \wedge b$
<code>_mm_andnot_ps</code> <code>_mm_andnot_pd</code>	<code>__m128 a, __m128 b</code>	<code>__m128</code>	$r = \sim a \& b$

Таблиця А.4 – Функції порівняння

Ім'я функції	Параметри	Результат	Призначення
--------------	-----------	-----------	-------------

<code>_mm_cmp<умова>_ss</code>	<code>__m128 a, __m128 b</code>	<code>__m128</code>	<code>r[0]= (a[0] β b[0]) ? 0xffffffff : 0</code> <code>r[i]= a[i] (i = 1..3)</code>
<code>_mm_cmp<умова>_ps</code>	<code>__m128 a, __m128 b</code>	<code>__m128</code>	<code>r[i]= (a[i] β b[i]) ? 0xffffffff : 0</code> <code>(i = 0..3)</code>
<code>_mm_cmp<умова>_sd</code>	<code>__m128d a, __m128d b</code>	<code>__m128d</code>	<code>r[0]= (a[0] β b[0]) ? 0xffffffff : 0</code> <code>r[i]= a[i] (i = 1)</code>
<code>_mm_cmp<умова>_pd</code>	<code>__m128d a, __m128d b</code>	<code>__m128d</code>	<code>r[i]= (a[i] β b[i]) ? 0xffffffff : 0</code> <code>(i = 0..1)</code>
<code>_mm_comi<умова>_ss</code> <code>_mm_ucomi<умова>_ss</code> <code>_mm_comi<умова>_sd</code> <code>_mm_ucomi<умова>_sd</code>	<code>__m128d a, __m128d b</code>	<code>__m128d</code>	<code>(a[0] β b[0])? 1:0</code>

β - умова порівняння

Таблиця А.5 – Функції для перетворення даних. Дані типу float/ double

Ім'я функції	Параметри	Результат	Призначення
<code>_mm_cvtss_si32</code> <code>_mm_cvtss_si32</code>	<code>__m128 a</code>	<code>int</code>	<code>r = a [0]</code>
<code>_mm_cvtps_pi32</code> <code>_mm_cvtps_pi32</code>	<code>__m128 a</code>	<code>__m64</code>	<code>r[i] = a [i]</code> <code>(i = 0..1)</code>
<code>_mm_cvtsi32_ss</code>	<code>__m128 a, int b</code>	<code>__m128</code>	<code>r[0]= (float)b</code> <code>r[i]= a[i]; (i = 1.. 3)</code>
<code>_mm_cvtsi64_ss</code>	<code>__m128 a, __int64 b</code>	<code>__m128</code>	<code>r[0]= (float)b</code> <code>r[i]= a[i]; (i = 1.. 3)</code>
<code>_mm_cvtpi32_ps</code>	<code>__m128 a, __m64 b</code>	<code>__m128</code>	<code>r[0]= (float)b[0]</code> <code>r[1]= (float)b[1]</code> <code>r[2]= a[2]; r[3]= a[3];</code>
<code>_mm_cvtpi16_ps</code>	<code>__m128 a, __m64 b</code>	<code>__m128</code>	<code>r[i]= (float)b[i]</code> <code>(i = 0.. 3) (b [i] – short)</code>
<code>_mm_cvtpu16_ps</code>	<code>__m128 a, __m64 b</code>	<code>__m128</code>	<code>r[i]= (float)b[i]</code> <code>(i = 0.. 3)(b [i] - ushort)</code>
<code>_mm_cvtpi8_ps</code>	<code>__m128 a, __m64 b</code>	<code>__m128</code>	<code>r[i]= (float)b[i]</code> <code>(i = 0.. 3) (b [i] -char)</code>
<code>_mm_cvtpu8_ps</code>	<code>__m128 a, __m64 b</code>	<code>__m128</code>	<code>r[i]= (float)b[i]</code> <code>(i = 0.. 3) (b [i] - uchar)</code>
<code>_mm_cvtpi32x2_ps</code>	<code>__m64 a, __m64 b</code>	<code>__m128</code>	<code>r[i]= (float)a[i](i = 0.. 3)</code>
<code>_mm_cvtps_pi16</code>	<code>__m128 a</code>	<code>__m64</code>	<code>r[i]=(short)a[i](i = 0.. 3)</code>
<code>_mm_cvtps_pi8</code>	<code>__m128 a</code>	<code>__m64</code>	<code>r[i]= (char)a[i],(i= 0.. 3)</code>
<code>_mm_cvtpd_ps</code>	<code>__m128d</code>	<code>__m128</code>	<code>r[0] = (float) a[0]</code> <code>r[1] = (float) a[1]</code> <code>r[2] = r[3] = 0</code>
<code>_mm_cvtps_pd</code>	<code>__m128</code>	<code>__m128d</code>	<code>r[0]=(double) a[0]</code> <code>r[1] = (double) a[1]</code>
<code>_mm_cvtepi32_pd</code>	<code>__m128i</code>	<code>__m128d</code>	<code>r[0]=(double) a[0]</code> <code>r[1] = (double) a[1]</code>

Ім'я функції	Параметри	Результат	Призначення
<code>_mm_cvtpd_epi32</code> <code>_mm_cvttpd_epi32</code>	<code>__m128d</code>	<code>__m128i</code>	<code>r[0] = (int) a[0]</code> <code>r[1] = (int) a[1]</code> <code>r[2] = r[3] = 0</code> Округлення/ Усікання
<code>_mm_cvtsd_si32</code> <code>_mm_cvttssd_si32</code>	<code>__m128d</code>	<code>int</code>	<code>r[0] = (int) a[0]</code> Округлення/ Усікання
<code>_mm_cvtsd_ss</code>	<code>__m128 a, __m128d b</code>	<code>__m128</code>	<code>r[0] = (float) b[0]</code> <code>r[1] = a[1];</code> <code>r[2] = a[2];</code> <code>r[3] = a[3]</code>
<code>_mm_cvtsi32_sd</code>	<code>__m128d a, int b</code>	<code>__m128d</code>	<code>r[0] = (double) b</code> <code>r[1] = a[1]</code>
<code>_mm_cvtss_sd</code>	<code>__m128d a, __m128 b</code>	<code>__m128d</code>	<code>r[0] = (double) b[0]</code> <code>r[1] = a[1]</code>
<code>_mm_cvtepi32_ps</code>	<code>__m128i</code>	<code>__m128</code>	<code>r[i] = (float) a[i]</code> (<code>i = 0..3</code>)
<code>_mm_cvtps_epi32</code> <code>_mm_cvttps_epi32</code>	<code>__m128 a</code>	<code>__m128i</code>	<code>r[i] = (int) a[i]</code> (<code>i = 0..3</code>) Округлення/ Усікання
<code>_mm_cvtpd_pi32</code> <code>_mm_cvttpd_pi32</code>	<code>A</code>	<code>__m64</code>	<code>r[i] = (int) a[i]</code> (<code>i = 0, 1</code>) Округлення/ Усікання
<code>_mm_cvtpi32_pd</code>	<code>__m64 a</code>	<code>__m128d</code>	<code>r[i] = (double) a[i]</code> (<code>i = 0, 1</code>)

Таблиця А.6 – Функції для округлення, SSE4

Ім'я функції	Параметри	Результат	Призначення
<code>_mm_round_ss</code>	<code>__m128 a,</code> <code>__m128 b,</code> <code>const int cntrl</code>	<code>__m128</code>	<code>r[0] = rnd (b[0])</code> <code>r[i] = a[i]</code> <code>i = 1..3</code>
<code>_mm_round_ps</code>	<code>__m128 a,</code> <code>const int cntrl</code>	<code>__m128</code>	<code>r[i] = rnd (a[i])</code> <code>i = 0..3</code>
<code>_mm_round_sd</code>	<code>__m128d a,</code> <code>__m128d b,</code> <code>const int cntrl</code>	<code>__m128d</code>	<code>r[0] = rnd (b[0])</code> <code>r[1] = a[1]</code>
<code>_mm_round_pd</code>	<code>__m128d a,</code> <code>const int cntrl</code>	<code>__m128</code>	<code>r[i] = rnd (a[i])</code> <code>i = 0..1</code>

Таблиця А.7 – Функції для керування кешем

Ім'я функції	Параметри	Результат	Призначення
<code>_mm_stream_ps</code>	<code>float * p, __m128 a</code>		<code>*p = a</code> (<code>p</code> - вирівняний)
<code>_mm_stream_ss</code>	<code>float * p, __m128 a</code>		<code>*p = a[0]</code> (<code>p</code> - вирівняний)
<code>_mm_stream_sd</code>	<code>double * p, __m128d a</code>		<code>*p = a[0]</code> (<code>p</code> - вирівняний)
<code>_mm_stream_pd</code> (SSE2)	<code>double * p, __m128 a</code>		<code>*p = a</code> (<code>p</code> - вирівняний)

Таблиця А.8 – Функції для ініціалізації та завантаження блоку даних з масиву

Ім'я функції	Параметри	Результат	Призначення
<code>_mm_load_ss</code>	<code>float * p</code>	<code>__m128</code>	<code>r[0] = *p; r[i] = 0 (i = 1..3)</code>

Ім'я функції	Параметри	Результат	Призначення
<code>_mm_load1_ps</code> <code>_mm_load_ps1</code>	<code>float * p</code>	<code>__m128</code>	<code>r[i] = *p;</code>
<code>_mm_load_ps</code> <code>_mm_loadu_ps</code>	<code>float * p</code>	<code>__m128</code>	<code>r[i] = p[i]; (i = 0..3)</code>
<code>_mm_loadr_ps</code>	<code>float * p</code>	<code>__m128</code>	<code>r[i] = p[3 - i]; (i = 0..3)</code>
<code>_mm_loadh_pi</code>	<code>__m128 a, __m64 * p</code>	<code>__m128</code>	<code>r[0]=a[0]; r[1]=a[1] r[2]=p[0]; r[3]=p[1]</code>
<code>_mm_loadl_pi</code>	<code>__m128 a, __m64 * p</code>	<code>__m128</code>	<code>r[0]=p[0]; r[1]=p[1] r[2]=a[2]; r[3]=a[3]</code>
<code>_mm_loadl_ps</code>	<code>__m128 a, float *p</code>	<code>__m128</code>	<code>r[0] = p[0]; r[1] = p[1] r[2] = a[2]; r[3] = a[3]</code>
<code>_mm_loadh_ps</code>	<code>__m128 a, float *p</code>	<code>__m128</code>	<code>r[0] = a[0]; r[1] = a[1] r[2] = p[0]; r[3] = p[1]</code>
<code>_mm_store_ss</code>	<code>float * p, __m128 a</code>	немає	<code>p[0]= a[0]</code>
<code>_mm_storeh_pi</code>	<code>__m64 * p, __m128 a</code>	Немає	<code>p[0] = a[2]; p[1] = a[3]</code>
<code>_mm_storel_pi</code>	<code>__m64 * p, __m128 a</code>	Немає	<code>p[0] = a[0]; p[1] = a[1]</code>
<code>_mm_storer_ps</code>	<code>Float * p, __m128 a</code>	Немає	<code>p[i] = a[3 - i]</code>
<code>_mm_store_ps</code> <code>_mm_storeu_ps</code>	<code>float * p, __m128 a</code>	Немає	<code>p[i] = a[i]; (p - не вирівняний)</code>
<code>_mm_store1_ps</code> <code>_mm_store_ps1</code>	<code>float * p, __m128 a</code>	немає	<code>p[i]= a[0]; i = 0..3</code>
<code>_mm_store_pd</code> <code>_mm_storeu_pd</code>	<code>float * p, __m128 a</code>	Немає	<code>p[i] = a[i]) (i = 0..3)</code>
<code>_mm_storer_pd</code>	<code>double * p, __m128 a</code>	Немає	<code>p[i] = a[1 - i]) (i = 0..1)</code>
<code>_mm_set_ss</code>	<code>float a</code>	<code>__m128</code>	<code>r[0]=a; r[i]=0(i=1..3)</code>
<code>_mm_set1_ps</code> <code>_mm_set_ps1</code>	<code>float a</code>	<code>__m128</code>	<code>r[i] = a; (i= 0..3)</code>
<code>_mm_set_ps</code>	<code>float z, float y, float x, float w</code>	<code>__m128</code>	<code>r[0] = w; r[1] = x; r[2] = y; r[3] = z</code>
<code>_mm_setr_ps</code>	<code>float z, float y, float x, float w</code>	<code>__m128</code>	<code>r[0] = z; r[1] = y; r[2] = x; r[3] = w</code>
<code>_mm_setzero_ps</code>	Немає	<code>__m128</code>	<code>r = 0</code>
<code>_mm_move_sd</code>	<code>__m128 a, __m128 b</code>	<code>__m128</code>	<code>r[0]= b[0]; r[1]= a[1]</code>
<code>_mm_move_ss</code>	<code>__m128 a, __m128 b</code>	<code>__m128</code>	<code>r[0]= b[0]; r[1]= a[1]</code>

Таблиця А.9 – Спеціальні функції

Ім'я функції	Параметри	Результат	Призначення
<code>_mm_shuffle_ps</code>	<code>__m128 a, __m128 b, int mask</code>	<code>__m128</code>	<code>r[0] = a[w]; r [1] = a [x]; r[2] = b[y]; r[3] = b [z]</code>
<code>_mm_shuffle_pd</code>	<code>__m128d a, __m128d b, int mask</code>	<code>__m128</code>	<code>r[0] = a[y]; r [1] = b [x];</code>
<code>_mm_unpackhi_ps</code>	<code>__m128 a, __m128 b</code>	<code>__m128</code>	<code>r[0]= a[2] r[1]= b[2] r[2]= a[3] r[3]= b[3]</code>
<code>_mm_unpackhi_pd</code>	<code>__m128d a, __m128d b</code>	<code>__m128d</code>	<code>r[0]= a[1] r[1]= b[1]</code>
<code>_mm_unpacklo_ps</code>	<code>__m128 a, __m128 b</code>	<code>__m128</code>	<code>r[0]= a[0] r[1]= b[0]</code>

Ім'я функції	Параметри	Результат	Призначення
			$r[2] = a[1]$ $r[3] = b[1]$
<code>__mm_unpacklo_pd</code>	<code>__m128d a, __m128d b</code>	<code>__m128d</code>	$r[0] = a[0]$ $r[1] = b[0]$
<code>__mm_movehl_ps</code>	<code>__m128 a, __m128 b</code>	<code>__m128</code>	$r[3] = a[3]$ $r[2] = a[2]$ $r[1] = b[3]$ $r[0] = b[2]$
<code>__mm_movehl_pd</code>	<code>__m128d a, __m128d b</code>	<code>__m128d</code>	$r[1] = a[1]$ $r[0] = b[1]$
<code>__mm_movelh_ps</code>	<code>__m128 a, __m128 b</code>	<code>__m128</code>	$r[3] = b[1]$ $r[2] = b[0]$ $r[1] = a[1]$ $r[0] = a[0]$
<code>__mm_movelh_pd</code>	<code>__m128d a, __m128d b</code>	<code>__m128d</code>	$r[1] = b[0]$ $r[0] = a[0]$
<code>__mm_movemask_ps</code>	<code>__m128 a</code>	int	$r = ((\text{sign}(a[3])) \ll 3) $ $((\text{sign}^3(a[2])) \ll 2) $ $((\text{sign}(a[1])) \ll 1) $ $((\text{sign}(a[0])))$
<code>__mm_movemask_pd</code>	<code>__m128d a</code>	int	$r = ((\text{sign}(a[1])) \ll 1) $ $((\text{sign}(a[0])))$

ДОДАТОК Б

SSE функції для роботи з цілими числами

Таблиця Б.1 – Функції завантаження

Ім'я функції	Параметри	Результат	Призначення
<code>__mm_load_si128</code>	<code>__m128i const*</code>	<code>__m128i</code>	$r = *pa$
<code>__mm_loadu_si128</code>	<code>__m128i const*</code>	<code>__m128i</code>	$r = *pa$
<code>__mm_lddqu_si128</code> (SSE3)	<code>__m128i const* pa</code>	<code>__m128i</code>	$r = *pa$; Якщо адреса вирівняна, то просто копіює
<code>__mm_loadl_epi64</code>	<code>__m128i const*</code>	<code>__m128i</code>	$r[0] = pa[0]; r[1] = 0;$
<code>__mm_stream_si128</code>	<code>__m128i* pa, __m128i b</code>		$*pa = b$ (без використання проміжних кешів)
<code>__mm_stream_load_si128</code> (SSE4)	<code>__m128i* pa</code>	<code>__m128i</code>	$r = *pa$

³ Sign – операція визначення знаку числа. Результат дорівнює 1 для негативних чисел і 0 для позитивних (знаковий біт числа)

Ім'я функції	Параметри	Результат	Призначення
<code>_mm_store_si128</code>	<code>__m128i *p, __m128i a</code>		<code>*p = a;</code>
<code>_mm_storeu_si128</code>	<code>__m128i *p, __m128i a</code>		<code>*p = a;</code>
<code>_mm_storel_epi64</code>	<code>__m128i *p, __m128i a</code>		<code>p[0] = a [0];</code>
<code>_mm_move_epi64</code>	<code>__m128i a</code>	<code>__m128i</code>	
<code>_mm_movpi64_epi64</code>	<code>__m64</code>	<code>__m128i</code>	<code>r[0] = a; r[1] = 0</code>
<code>_mm_movepi64_pi64</code>	<code>__m128i a</code>	<code>__m64</code>	<code>r = a[0]</code>

Таблиця В.2 – Арифметичні функції

Ім'я функції	Параметри	Результат	Призначення
<code>_mm_add_epi8</code> <code>_mm_add_epi16</code> <code>_mm_add_epi32</code> <code>_mm_add_epi64</code>	<code>__m128i a, __m128i b</code>	<code>__m128i</code>	<code>r[i]=a[i]+b[i]; (i=0..15);</code> <code>r[i]=a[i]+b[i]; (i=0..7);</code> <code>r[i]=a[i]+b[i]; (i=0..3);</code> <code>r[i]=a[i]+b[i]; (i=0..1);</code>
<code>_mm_adds_epi8</code> <code>_mm_adds_epi16</code>	<code>__m128i a, __m128i b</code>	<code>__m128i</code> , Насичення	<code>r[i]=a[i]+b[i]; (i=0..15);</code> <code>r[i]=a[i]+b[i]; (i=0..7);</code>
<code>_mm_adds_epu8</code> <code>_mm_adds_epu16</code>	<code>__m128i a, __m128i b</code>	<code>__m128i</code> Насичення	<code>r[i]=a[i]+b[i]; (i=0..15);</code> <code>r[i]=a[i]+b[i]; (i=0..7);</code>
<code>_mm_hadd_epi16</code> <code>_mm_hadds_epi16</code>	<code>__m128i a, __m128i b</code> (ssse3)	<code>__m128i</code>	<code>r[i]=a[2i]+a[2i+1]; (i=0..3);</code> <code>r[i+4]=b[2i]+b[2i+1]; i=0..3;</code>
<code>_mm_hadd_epi32</code> (ssse3)	<code>__m128i a, __m128i b</code>	<code>__m128i</code>	<code>r[i]=a[2i]+a[2i+1]; (i=0..1);</code> <code>r[i+2]=b[2i]+b[2i+1]; (i=0,1)</code>
<code>_mm_avg_epu8</code> <code>_mm_avg_epu16</code>	<code>__m128i a, __m128i b</code>	<code>__m128i</code>	<code>r[i]=(a[i]+b[i]) / 2; (i=0..15);</code> <code>r[i]=(a[i]+b[i]) / 2; (i=0..7);</code>
<code>_mm_sub_epi8</code> <code>_mm_sub_epi16</code> <code>_mm_sub_epi32</code> <code>_mm_sub_epi64</code>	<code>__m128i a, __m128i b</code>	<code>__m128i</code>	<code>r[i]=a[i]-b[i]; (i=0..15);</code> <code>r[i]=a[i]-b[i]; (i=0..7);</code> <code>r[i]=a[i]-b[i]; (i=0..3);</code> <code>r[i]=a[i]-b[i]; (i=0..1);</code>
<code>_mm_subs_epi8</code> <code>_mm_subs_epi16</code>	<code>__m128i a, __m128i b</code>	<code>__m128i</code>	<code>r[i]=a[i]-b[i]; (i=0..15);</code> <code>r[i]=a[i]-b[i]; (i=0..7);</code>
<code>_mm_subs_epu8</code> <code>_mm_subs_epu16</code>	<code>__m128i a, __m128i b</code>	<code>__m128i</code>	<code>r[i]=a[i]-b[i]; (i=0..15);</code> <code>r[i]=a[i]-b[i]; (i=0..7);</code>
<code>_mm_hsub_epi16</code> <code>_mm_hsubs_epi16</code> (SSSE3)	<code>__m128i a, __m128i b</code>	<code>__m128i</code>	<code>r[i]=a[2i]-a[2i+1]; (i=0..3);</code> <code>r[i+4]=b[2i]-b[2i+1];</code> <code>(i=0..3);</code>
<code>_mm_hsub_epi32</code> (SSSE3)	<code>__m128i a, __m128i b</code>	<code>__m128i</code>	<code>r[i]=a[2i]-a[2i+1]; (i=0..1);</code> <code>r[i+2]=b[2i]-b[2i+1]; (i=0..1)</code>
<code>_mm_sad_epu8</code>	<code>__m128i a, __m128i b</code>	<code>__m128i</code>	<code>r[0]=∑ abs (a[i] - b[i]);</code> <code>i = 0..7; r[i]=0 (i = 1..3)</code> <code>r[4]=∑ abs (a[8+i] -b[8 +i]);</code> <code>i = 0..7;</code> <code>r[i]=0 (i = 5..7)</code>
<code>_mm_madd_epi16;</code>	<code>__m128i a, __m128i b</code>	<code>__m128i</code>	<code>r[i]=(a[2i] * b[2i]) +</code> <code>(a[2i+1] * b[2i+1]); (i = 0..3)</code>
<code>_mm_maddubs_epi16</code> (SSSE3)	<code>__m128i a, __m128i b</code>	<code>__m128i</code>	<code>r[i]=sat (a[2i] * b[2i] +</code> <code>a[2i+1] * b[2i+1]) (i = 0..7)</code>
<code>_mm_mulhi_epi16</code>	<code>__m128i a, __m128i b</code>	<code>__m128i</code>	<code>r[i]=high(a[i] * b[i]); i = 0, 7</code>
<code>_mm_mullo_epi16</code> <code>_mm_mullo_epi32</code> (SSE4)	<code>__m128i a, __m128i b</code>	<code>__m128i</code>	<code>r[i]=low(a[i] * b[i]); i= 0,7</code> <code>r[i]=low(a[i] * b[i]); i = 0, 3,</code>
<code>_mm_mulhi_epu16</code>	<code>__m128i a, __m128i b</code>	<code>__m128i</code>	<code>r[i]=high(a[i] * b[i]); i = 0,7</code>

Ім'я функції	Параметри	Результат	Призначення
<code>_mm_mul_epi32</code> <code>_mm_mul_epu32</code> (SSE4)	<code>__m128i a, __m128i b</code>	<code>__m128i</code>	<code>r[0] = low(a[0] * b[0]);</code> <code>r[1] = high(a[0] * b[0]);</code> <code>r[2] = low(a[2] * b[2]);</code> <code>r[3] = high(a[2] * b[2]);</code>
<code>_mm_mulhrs_epi16</code>	<code>__m128i a, __m128i b</code>	<code>__m128i</code>	<code>r[i]=short((((a[i]*b[i])+</code> <code>0x4000) >> 15)</code>
<code>_mm_min_epi8(16,32)</code> <code>_mm_max_epi8(16,32)</code> <code>_mm_min_epu8(16,32)</code> <code>_mm_max_epu8(16,32)</code>	<code>__m128i a, __m128i b</code> (SSE4)	<code>__m128i</code>	<code>r[i]=min(a[i], b[i]);</code> <code>r[i]=max(a[i], b[i]);</code> (int, unsigned)

Таблиця Б.3 – Функції для роботи з бітами

Ім'я функції	Параметри	Результат	Призначення
<code>_mm_xor_si128</code>	<code>__m128i a,</code> <code>__m128i b</code>	<code>__m128i</code>	<code>r = a ^ b</code>
<code>_mm_and_si128</code>	<code>__m128i a,</code> <code>__m128i b</code>	<code>__m128i</code>	<code>r = a & b</code>
<code>_mm_or_si128</code>	<code>__m128i a,</code> <code>__m128i b</code>	<code>__m128i</code>	<code>r = a b</code>
<code>_mm_andnot_si128</code>	<code>__m128i a,</code> <code>__m128i b</code>	<code>__m128i</code>	<code>r = a & ~b</code>
<code>_mm_slli_si128</code>	<code>__m128i a,</code> <code>int c</code>	<code>__m128i</code>	<code>r = a << c</code>
<code>_mm_slli_epi16</code>	<code>__m128i a,</code> <code>int c</code>	<code>__m128i</code>	<code>r[i] = a[i] << c; i = 0..7</code>
<code>_mm_sll_epi16</code>	<code>__m128i a,</code> <code>__m128i b</code>	<code>__m128i</code>	<code>r[i] = a[i] << b[0]; i = 0..7</code>
<code>_mm_slli_epi32</code>	<code>__m128i a,</code> <code>int c</code>	<code>__m128i</code>	<code>r[i] = a[i] << c; i = 0..3</code>
<code>_mm_sll_epi32</code>	<code>__m128i a,</code> <code>__m128i b</code>	<code>__m128i</code>	<code>r[i] = a[i] << b[0]; i = 0..3</code>
<code>_mm_slli_epi64</code>	<code>__m128i a,</code> <code>int c</code>	<code>__m128i</code>	<code>r[i] = a[i] << c; i = 0..1</code>
<code>_mm_sll_epi64</code>	<code>__m128i a,</code> <code>__m128i b</code>	<code>__m128i</code>	<code>r[i] = a[i] << b[0]; i = 0..1</code>
<code>_mm_srai_epi16</code>	<code>__m128i a,</code> <code>int c</code>	<code>__m128i</code>	<code>r[i] = a[i] >> c; i = 0..7</code> (int)
<code>_mm_sra_epi16</code>	<code>__m128i a,</code> <code>__m128i b</code>	<code>__m128i</code>	<code>r[i] = a[i] >> b[0]; i = 0..7</code> (int)
<code>_mm_srai_epi32</code>	<code>__m128i a,</code> <code>int c</code>	<code>__m128i</code>	<code>r[i] = a[i] >> c; i = 0..3</code> (int)
<code>_mm_sra_epi32</code>	<code>__m128i a,</code> <code>__m128i b</code>	<code>__m128i</code>	<code>r[i] = a[i] >> b[0]; i = 0..3</code> (int)
<code>_mm_srli_si128</code>	<code>__m128i a,</code> <code>int c</code>	<code>__m128i</code>	<code>r = a >> c; (unsigned)</code>
<code>_mm_srli_epi16</code>	<code>__m128i a,</code> <code>int c</code>	<code>__m128i</code>	<code>r[i] = a[i] >> c; i = 0..7</code> (unsigned)
<code>_mm_srl_epi16</code>	<code>__m128i a,</code> <code>__m128i b</code>	<code>__m128i</code>	<code>r[i] = a[i] >> b[0]; i = 0..7</code> (unsigned)

Ім'я функції	Параметри	Результат	Призначення
<code>_mm_srli_epi32</code>	<code>__m128i a</code> , <code>int c</code>	<code>__m128i</code>	<code>r[i] = a[i] >> c; i = 0..3</code> (unsigned)
<code>_mm_srl_epi32</code>	<code>__m128i a</code> , <code>__m128i b</code>	<code>__m128i</code>	<code>r[i] = a[i] >> b[0]; i = 0..3</code> (unsigned)
<code>_mm_srli_epi64</code>	<code>__m128i a</code> , <code>int c</code>	<code>__m128i</code>	<code>r[i] = a[i] >> c; i = 0..1</code> (unsigned)
<code>_mm_srli_epi64</code>	<code>__m128i a</code> , <code>__m128i b</code>	<code>__m128i</code>	<code>r[i] = a[i] >> b[0]; i = 0..1</code> (unsigned)

Таблиця Б.4 – Функції порівняння даних

Ім'я функції	Параметри	Результат	Призначення
<code>_mm_cmp<Умова>_epi8</code>	<code>__m128i a</code> , <code>__m128i b</code>	<code>__m128i</code>	<code>r[i] = a[i] q b[i]? -1:0</code> (<code>i = 0..15</code>)
<code>_mm_cmp<Умова>_epi16</code>	<code>__m128i a</code> , <code>__m128i b</code>	<code>__m128i</code>	<code>r[i] = a[i] q b[i]? -1:0</code> (<code>i = 0..7</code>)
<code>_mm_cmp<Умова>_epi32</code>	<code>__m128i a</code> , <code>__m128i b</code>	<code>__m128i</code>	<code>r[i] = a[i] q b[i]? -1:0</code> (<code>i = 0..3</code>)
<code>_mm_cmp<Умова>_epi64 (SSE4)</code>	<code>__m128i a</code> , <code>__m128i b</code>	<code>__m128i</code>	<code>r[i] = a[i] q b[i]? -1:0</code>
<code>_mm_testz_si128 SSE4</code>	<code>__m128i a</code> , <code>__m128i b</code>	<code>int</code>	<code>r = (a & b) == 0</code>
<code>_mm_testc_si128 SSE4</code>	<code>__m128i a</code> , <code>__m128i b</code>	<code>int</code>	<code>r = (a == b)? 1:0</code>
<code>_mm_testnzc_si128 SSE4</code>	<code>__m128i a</code> , <code>__m128i b</code>	<code>int</code>	<code>ZF = (a & b) == 0</code> <code>CF = (~a & b) == 0</code> <code>r = ~ZF & ~CF</code>

Таблиця Б.5 – Функції для упакування та розпакування

Ім'я функції	Параметри	Результат	Призначення
<code>_mm_packs_epi16</code>	<code>__m128i a</code> , (short) <code>__m128i b</code> (short)	<code>__m128i</code> (char)	<code>r[i] = sat(a[i]);</code> <code>r[8+i] = sat(b[i]);</code> <code>i = 0..7</code>
<code>_mm_packs_epi32</code>	<code>__m128i a</code> , (int) <code>__m128i b</code> (int)	<code>__m128i</code> (short)	<code>r[i] = sat(a[i]);</code> <code>r[4+i] = sat(b[i]);</code> <code>i = 0..3</code>
<code>_mm_packus_epi16</code>	<code>__m128i a</code> , (short) <code>__m128i b</code> (short)	<code>__m128i</code> (char)	<code>r[i] = sat(a[i]);</code> <code>r[8+i] = usat(b[i]);</code> <code>i = 0..7</code>
<code>_mm_unpackhi_epi8</code> <code>_mm_unpackhi_epi16</code> <code>_mm_unpackhi_epi32</code> <code>_mm_unpackhi_epi64</code>	<code>__m128i a</code> , <code>__m128i b</code>	<code>__m128i</code>	На парні місця ставляться старші блоки <i>a</i> . На непарні <i>b</i>
<code>_mm_unpacklo_epi8</code> <code>_mm_unpacklo_epi16</code> <code>_mm_unpacklo_epi32</code> <code>_mm_unpacklo_epi64</code>	<code>__m128i a</code> , <code>__m128i b</code>	<code>__m128i</code>	На парні місця ставляться молодші блоки <i>a</i> . На непарні <i>b</i>

Таблиця Б.6 – Функції для ініціалізації даних

Ім'я функції	Параметри	Результат	Призначення
<code>_mm_set_epi64</code> <code>_mm_setr_epi64</code>	<code>__m64 a1, __m64 a0</code>	<code>__m128i</code>	<code>r[0] = a0; r[1] = a1;</code> <code>r[0] = a1; r[1] = a0;</code>
<code>_mm_set_epi32</code> <code>_mm_setr_epi32</code>	<code>int a3, int a2, int a1,</code> <code>int a0</code>	<code>__m128i</code>	<code>r[0] = a0; ... r[3] = a3;</code> <code>r[0] = a3; ... r[3] = a0;</code>
<code>_mm_set_epi16</code> <code>_mm_setr_epi16</code>	<code>short a7, short a6, ...,</code> <code>short a0</code>	<code>__m128i</code>	<code>r[0] = a0; ... r[7] = a7;</code> <code>r[0] = a7; ... r[7] = a0;</code>
<code>_mm_set_epi8</code> <code>_mm_setr_epi8</code>	<code>char a15, char a14, ...,</code> <code>char a0</code>	<code>__m128i</code>	<code>r[0] = a0; ... r[15] = a15;</code> <code>r[0] = a15; ... r[15] = a0;</code>
<code>_mm_set1_epi64</code>	<code>__m64 a,</code>	<code>__m128i</code>	<code>r[i] = a; (i = 0, 1)</code>
<code>_mm_set1_epi16</code>	<code>short a</code>	<code>__m128i</code>	<code>r[i] = a; (i = 0, 7)</code>
<code>_mm_set1_epi8</code>	<code>char a</code>	<code>__m128i</code>	<code>r[i] = a; (i = 0, 15)</code>
<code>_mm_setzero_si128</code>		<code>__m128i</code>	<code>r = 0</code>

Таблиця Б.7 – Функції для перемішування даних

Ім'я функції	Параметри	Результат	Призначення
<code>_mm_shufflehi_epi16</code>	<code>__m128i a,</code> <code>int mask</code>		У старших 4 півсловах зміна по масці. <code>_MM_SHUFFLE(z, y, x, w)</code>
<code>_mm_shufflelo_epi16</code>	<code>__m128i a,</code> <code>int mask</code>		У молодших 4 півсловах зміна по масці <code>_MM_SHUFFLE(z, y, x, w)</code>
<code>_mm_shuffle_epi32</code>	<code>__m128i a,</code> <code>int mask</code>	<code>__m128i</code>	Зміна 4-х байтових слів по масці <code>_MM_SHUFFLE(z, y, x, w)</code>

Таблиця Б.8 – Функції для перетворення даних

Ім'я функції	Параметри	Результат	Призначення
<code>_mm_cvtsi32_si128</code>	<code>int a</code>	<code>__m128i</code>	<code>r[0] = a; r[i] = 0; (i=1..3)</code>
<code>_mm_cvtsi128_si32</code>	<code>__m128i</code>	<code>int</code>	<code>r = a[0]</code>
<code>_mm_cvtepi8_epi16</code> (SSE4)	<code>__m128i a</code>	<code>__m128i</code>	<code>r[2*i]=a[i];</code> <code>r[2i+1]=a[i]<0?-1:0; (i=0..7)</code>
<code>_mm_cvtepu8_epi16</code> (SSE4)	<code>__m128i a</code> (short)	<code>__m128i</code> (char)	<code>r[2*i]=a[i];</code> <code>r[2i+1]= 0; (i=0..7)</code>
Ім'я функції	Параметри	Результат	Призначення
<code>_mm_cvtepi8_epi32</code> (SSE4)	<code>__m128i a</code>	<code>__m128i</code>	<code>r[4*i]=a[i];</code> <code>r[4i+j]=a[i]<0?-1:0;(i=0..3)</code> <code>(j=1..3)</code>
<code>_mm_cvtepu8_epi32</code> (SSE4)	<code>__m128i a</code>	<code>__m128i</code>	<code>r[4*i]=a[i];</code> <code>r[4i+j]=0; (i=0..3) (j=1..3)</code>
<code>_mm_cvtepi8_epi64</code> (SSE4)	<code>__m128i a</code>	<code>__m128i</code>	<code>r[4*i]=a[i];</code> <code>r[4i+j]=a[i]<0?-1:0;(i=0..1)</code> <code>(j=1..7)</code>

Ім'я функції	Параметри	Результат	Призначення
<code>_mm_cvtepu8_epi64</code> (SSE4)	<code>__m128i a</code>	<code>__m128i</code>	<code>r[4*i]=a[i];</code> <code>r[4i+j]= 0; (i=0..1) (j=1..7)</code>
<code>_mm_cvtepi16_epi32</code> (SSE4)	<code>__m128i a</code>	<code>__m128i</code>	<code>r[2*i]=a[i];</code> <code>r[2i+1]=a[i]< 0? -1:0;</code> <code>(i=0..3)</code>
<code>_mm_cvtepu16_epi32</code> (SSE4)	<code>__m128i a</code>	<code>__m128i</code>	<code>r[2*i]=a[i];</code> <code>r[2i+1]= 0; (i=0..3)</code>
<code>_mm_cvtepi16_epi64</code> (SSE4)	<code>__m128i a</code>	<code>__m128i</code>	<code>r[4*i]=a[i];</code> <code>r[2i+j]=a[i]< 0? -1:0;</code> <code>(i=0..1) (j=1..3)</code>
<code>_mm_cvtepu16_epi64</code> (SSE4)	<code>__m128i a</code>	<code>__m128i</code>	<code>r[4*i]=a[i];</code> <code>r[2i+j]= 0; (i=0..1) (j=1..3)</code>
<code>_mm_cvtepi32_epi64</code> (SSE4)	<code>__m128i a</code>	<code>__m128i</code>	<code>r[2*i]=a[i];</code> <code>r[2i+1]=a[i]< 0? -1:0;</code> <code>(i=0..1)</code>
<code>_mm_cvtepu32_epi64</code> (SSE4)	<code>__m128i a</code>	<code>__m128i</code>	<code>r[2*i]=a[i];</code> <code>r[2i+1]= 0; (i=0..1)</code>

Таблиця Б.9 – Інші функції (дозволяють отримати окремий компонент з блоку даних, або переставити їх відповідно заданому порядку)

Ім'я функції	Параметри	Результат	Призначення
<code>_mm_movemask_epi8</code>	<code>(__m128i a</code>	Int	Знакові розряди кожного байта записуються в r
<code>_mm_maskmoveu_si128</code>	<code>__m128i a,</code> <code>__m128i b,</code> <code>char*p</code>	<code>__m128i</code>	<code>if (b[i] < 0)p[i] = a[i];</code> <code>(i = 0..15)</code>
<code>_mm_alignr_epi8</code> (SSSE3)	<code>__m128i a,</code> <code>__m128i b,</code> <code>Const int c</code>	<code>__m128i</code>	<code>temp = ab; r= temp >>(c*8)</code>
<code>_mm_blend_epi16</code> (sse4)	<code>__m128i a,</code> <code>__m128i b,</code> <code>Const int mask</code>	<code>__m128i</code>	<code>r[i]=(mask>>i)&1?</code> <code>a[i] : b[i] (i=0..7)</code> <code>(mask - 8 біт)</code>
<code>_mm_blendv_epi8</code> (sse4)	<code>__m128i a,</code> <code>__m128i b,</code> <code>__m128i mask</code>	<code>__m128i</code>	<code>r[i]=mask[i]&0x80?</code> <code>a[i] : b[0] (i=0..15)</code> <code>(mask - 16 байт)</code>
<code>_mm_insert_epi8</code> <code>_mm_insert_epi16</code> <code>_mm_insert_epi32, (sse4)</code>	<code>__m128i a,</code> <code>int b,</code> <code>const int ndx</code>	<code>__m128i</code>	<code>r[i]= (ndx == i) ? b : a[i]</code>
<code>_mm_extract_epi8</code> <code>_mm_extract_epi16</code> <code>_mm_extract_epi32 (sse4)</code>	<code>__m128i a,</code> <code>const int ndx</code>	Int	<code>if (ndx >max) ndx = max;</code> <code>r =a[ndx]</code>
<code>_mm_packus_epi32</code> (sse4)	<code>__m128i a,</code> <code>__m128i b</code> (int)	<code>__m128i</code>	<code>r[i]= (a[i] < 0) ? 0 :</code> <code>((a[i]>0xffff) ? 0xffff : a[i]) (i =</code> <code>0..3); r[4 +i]= (b[i] < 0) ? 0 :</code> <code>((b[i]>0xffff)?0xffff : b[i])</code> <code>(I = 0..3)</code>

Ім'я функції	Параметри	Результат	Призначення
<code>_mm_mpsadbw_epu8</code> (sse4)	<code>__m128i</code> <code>a, (uchar)</code> <code>__m128i</code> <code>b, (uchar)</code> <code>const int mask</code> (3 біти)	<code>__m128i</code> Short	<code>i = (mask >> 2) & 1 * 4</code> <code>j = (mask & 3) * 4</code> <code>for (k = 0; k < 8; k = k + 1) {</code> <code> t[0] = abs(a[i+k+0] - b[j+0])</code> <code> t[1] = abs(a[i+k+1] - b[j+1])</code> <code> t[2] = abs(a[i+k+2] - b[j+2])</code> <code> t[3] = abs(a[i+k+3] - b[j+3])</code> <code> r[k] = t[0] + t[1] + t[2] + t[3];</code> <code>}</code>
<code>_mm_extract_si64</code> (SSE4a)	<code>__m128i a,</code> <code>int Length,</code> <code>int Index</code>	<code>__m128i</code>	Виділення заданої кількості біт з <code>a</code>
<code>_mm_extract_si64</code> (SSE4a)	<code>__m128i a,</code> <code>__m128i b</code>	<code>__m128i</code>	Те ж, що попередня функція, але <code>b[0] = Length + (Index << 8)</code>
<code>_mm_insert_si64</code> (SSE4a)	<code>__m128i a,</code> <code>__m128i b</code> <code>int Length,</code> <code>int Index</code>	<code>__m128i</code>	Вставка в <code>a</code> бітів з <code>b</code>
<code>_mm_insert_si64</code> (SSE4a)	<code>__m128i a,</code> <code>__m128i b</code> <code>- Index</code>	<code>__m128i</code>	Вставка в <code>a</code> бітів з <code>b</code> . <code>Length = (b >> 72) % 64</code> <code>Index = (b >> 64) % 64</code>

ДОДАТОК В

AVX(AVX2) функції

Таблиця В.1 – Функції AVX(AVX2)⁴

Ім'я функції	Параметри	Результат	Призначення
<code>zeroall</code>	-		Значення всіх YMM регістрів = 0
<code>zeroupper</code>			значення всіх XMM регістрів залишається без змін, старша частина регістрів = 0
<code>broadcast_ss,</code> <code>broadcast_sd,</code> <code>broadcast_ps,</code> <code>broadcast_pd,</code> <code>_mm_broadcastss_ps,</code> <code>_mm_broadcastsd_pd,</code> <code>broadcastss_ps,</code> <code>broadcastsd_pd</code>	<code>float const *p</code> <code>double const *</code> <code>__m128 const *</code> <code>__m128d const *</code> <code>__m128</code> <code>__m128d</code> <code>__m128</code> <code>__m128d</code>	<code>__m256</code> <code>__m256d</code> <code>__m256</code> <code>__m256d</code> <code>__m128</code> <code>__m128d</code> <code>__m128</code> <code>__m128d</code>	Поширюють значення першого даного відповідного типу на усі компоненти блоку результату (дані з плаваючою крапкою)

⁴ Тут наведені тільки ті функції, які є в AVX(AVX2) і відсутні в SSE. Для скорочення імені функції, якщо вона починається з `_mm256_`, цей префікс не пишеться, наприклад функції `broadcast_ss` відповідає функція `_mm256_broadcast_ss`

__mm_broadcastb_epi8 __mm_broadcastw_epi16 __mm_broadcastd_epi32 __mm_broadcastq_epi64 broadcastb_epi8 broadcastw_epi16 broadcastd_epi32 broadcastq_epi64 broadcastsi128_si256	__m128i __m128i __m128i __m128i __m128i __m128i __m128i __m128i __m128i	__m128i __m128i __m128i __m128i __m256i __m256i __m256i __m256i __m256i	Поширюють значення першого даного відповідного типу на усі компоненти блоку результату (цілі дані)
maskload_ps maskload_pd maskload_epi32 maskload_epi64	float *, __m256i double*, __m256i int*, __m256i int64*, __m256i	__m256 __m256d __m256i __m256i	Завантаження в AVX результат значень, які відповідають полям маски, в яких 1 в старшому біті
maskstore_ps maskstore_pd maskstore_epi32 maskstore_epi64	float*, __m256i, __m256 double*, __m256i, __m256d int*, __m256i, __m256i int64*, __m256i, __m256i	void	Зберігання в пам'яті значень, які відповідають полям маски, в яких 1 в старшому біті
Ім'я функції	Параметри	Результат	Призначення

mask_i32gather_ps	__m256, float *, __m256i, __m256, int	__m256	Збирають блок з пам'яті, по заданим умовам. Перший параметр задає, звідки брати дані, якщо умова не виконується. Другий параметр задає початкову адресу пам'яті; Третій параметр (index), використовується разом з 5 параметром Четвертий параметр (маска). Копіювати, якщо, старший біт компонента 1. П'ятий параметр (scale) – розмір одного елементу. Фактичне зміщення в пам'яті дорівнює index * scale
mask_i32gather_pd	__m256d, double*, __m128i, __m256d, int	__m256d	
mask_i64gather_ps_mm256	__m128, float*, __m256i, __m128, int	__m128	
mask_i64gather_pd	__m256d, double*, __m256i, __m256d, int	__m256d	
mask_i32gather_epi32	__m256i, int *, __m256i, __m256i, int	__m256i	
mask_i32gather_epi64	__m256i, int64 *, __m128i, __m256i, int	__m256i	
mask_i64gather_epi32	__m128i, int*, __m256i, __m128i, int	__m128i	
mask_i64gather_epi64	__m256i, int64 *, __m256i, __m256i, int	__m256i	
_mm_mask_i32gather_epi32	__m128i, int*, __m128i, __m128i, int	__m128i	
_mm_mask_i32gather_epi64	__m128i, int64*, __m128i, __m128i, int	__m128i	
_mm_mask_i64gather_epi32	__m128i, int*, __m128i, __m128i, int	__m128i	
_mm_mask_i64gather_epi64	__m128i, int64*, __m128i, __m128i, int	__m128i	

i32gather_ps i32gather_pd ...			Функції відрізняються від попередніх відсутністю маски, копіювати треба в будь-якому разі, тому немає 1 і 4 параметрів
fmadd_ps fmsub_ps fnmadd_ps fnmsub_ps	__m256 a, __m256 b, __m256 c	__m256	$a[i]*b[i]+c[i]$, $i=0..7$ $a[i]*b[i]-c[i]$, $i=0..7$ $-a[i]*b[i]+c[i]$, $i=0..7$ $-a[i]*b[i]-c[i]$, $i=0..7$
fmadd_pd fmsub_pd fnmadd_pd fnmsub_pd	__m256d a, __m256d b, __m256d c	__m256	$a[i]*b[i]+c[i]$, $i=0..3$ $a[i]*b[i]-c[i]$, $i=0..3$ $-a[i]*b[i]+c[i]$, $i=0..3$ $-a[i]*b[i]-c[i]$, $i=0..3$
_mm_fmaddsub_ps _mm_fmsubadd_ps	__m128 a, __m128 b, __m128 c	__m128	$a[i]*b[i]-c[i]$ ($i = 2k + 1$, $k=0..3$); $a[i]*b[i]+c[i]$ ($i = 2k$, $k=0..3$)
_mm_fmaddsub_pd _mm_fmsubadd_pd	__m128d a, __m128d b, __m128d c	__m128d	$a[i]*b[i]-c[i]$ ($i = 2k + 1$, $k=0..1$); $a[i]*b[i]+c[i]$ ($i = 2k$, $k=0..1$)
testz_ps testc_ps testnzc_ps testz_pd testc_pd testnzc testz_si256 testnzc_si256 testc_si256	__m256 a, __m256 b __m256d a, __m256d b __m256i a, __m256i b	Int	$r = (a \& b) == 0 ? 1 : 0$ (flag Z) $r = \sim(a \& b) == 0 ? 1 : 0$ (flag C) $r = (!\text{flag Z}) \&\& (!\text{flag C}) ? 1 : 0$
cmp_ss cmp_ps cmp_sd cmp_pd	__m256 __m256 Int __m256d __m256d int	__m256 __m256d	Порівняння даних, які задані першими двома операндами. Останній параметр задає умову порівняння. Приклад умови: _CMP_NGT_US – не більше
extractf128_ps extractf128_pd extractf128_si256 extracti128_si256	__m256, int __m256d, int __m256i, int __m256i, int	__m128 __m128d __m128i __m128i	Виділення компоненту з заданим номером (номер задається другим параметром, може бути 0, або 1)
permute2f128_ps permute2f128_pd permute2f128_si256 permute2x128_si256	__m256, __m256, int __m256d, __m256d, int __m256i, __m256i, int	__m256 __m256d __m256i __m256i	Перемішування даних з 1 та 2 параметрів. Параметр 3 – маска – 8 бітів. 4 молодші біти для формування першої половини полів, старші 3 – для другої. Якщо старший біт відповідної тетрази 1 – відповідне половина дорівнює 0, якщо 0, то обирається з першого (молодші 2 біта =0, 1). Або другого операнду (2, 3)

permute4x64_pd permute4x64_epi64	__m256d, int __m256i, int	__m256d __m256i	Перемішуються 64-бітні компоненти блоку. Маска 8 бітів, задає номер для кожного з 4-х блоків результату
permutevar8x32_ps permutevar8x32_epi32	__m256, __m256i __m256i, __m256i	__m256 __m256i	Перемішуються 32-бітні компоненти блоку. Кожний компонент маски (другий параметр) задає місце відповідного компонента результату
alignr_epi8	__m256i, __m256i, int	__m256i	Молодші блоки gfhfvtnhsd об'єднуються в один блок, молодший блок першого даного становиться старшим блоком, а молодший блок другого – молодшим. Виконується їх зсув вправо на кількість байтів, які задаються константою (3 параметр). З отриманого результату залишається тільки 128 бітів, які записуються як молодший блок. Аналогічно формується старші 128 бітів результату