

组会报告

徐益

2018 年 7 月 29 日

1 工作内容

1. 完成 LDPC-mex-sim
2. 阅读 Intel® 64 and IA-32 Architectures Software Developer's Manual Volume 1
3. 学习 Intrinsics Guide 并测试

2 SIMD(single instruction, multiple data)

2.1 SIMD Execution Model

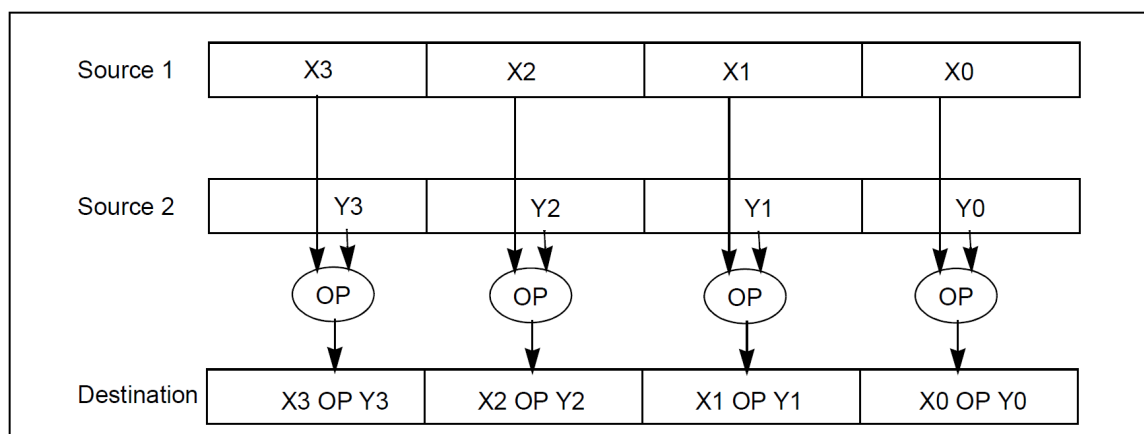


图 1: SIMD Execution Model

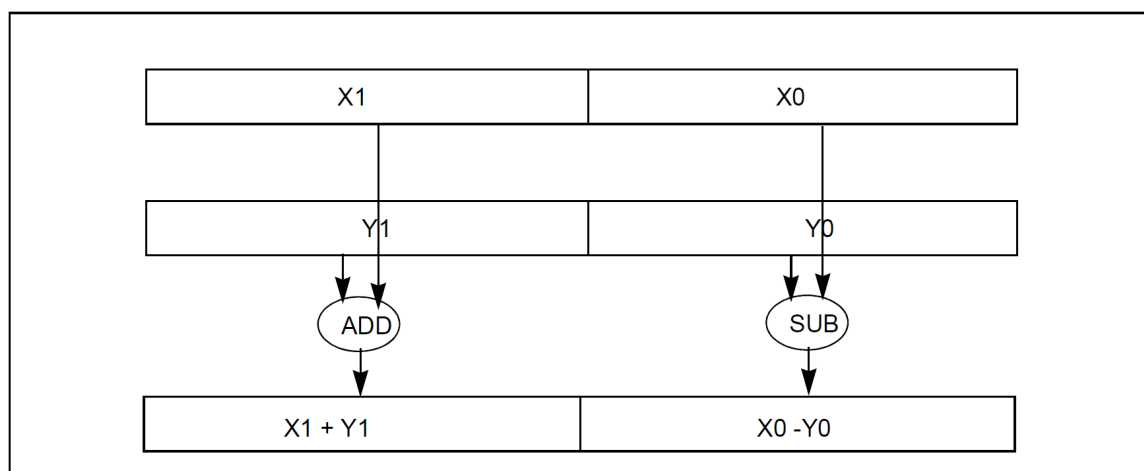


图 2: Asymmetric Processing in ADDSUBPD

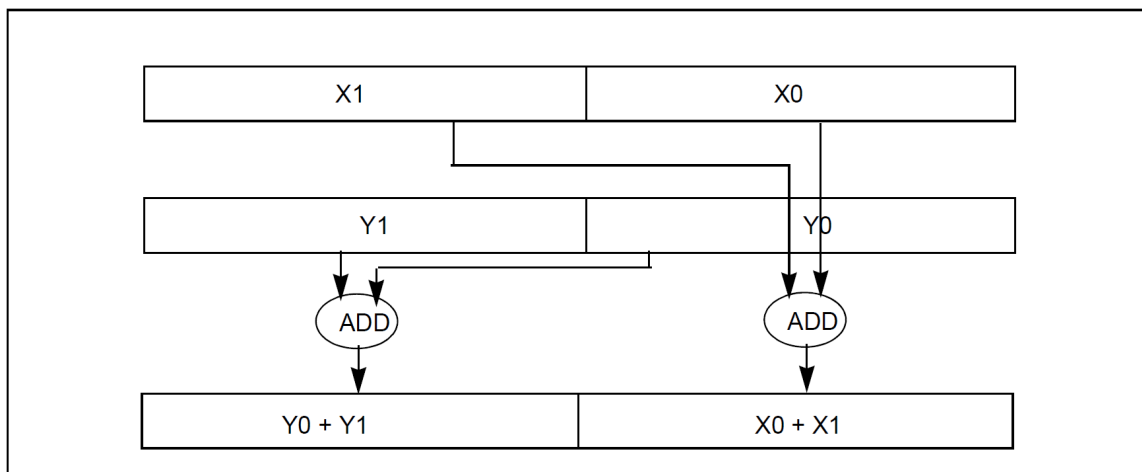


图 3: Horizontal Data Movement in HADDPD

2.2 SIMD Execution Environment

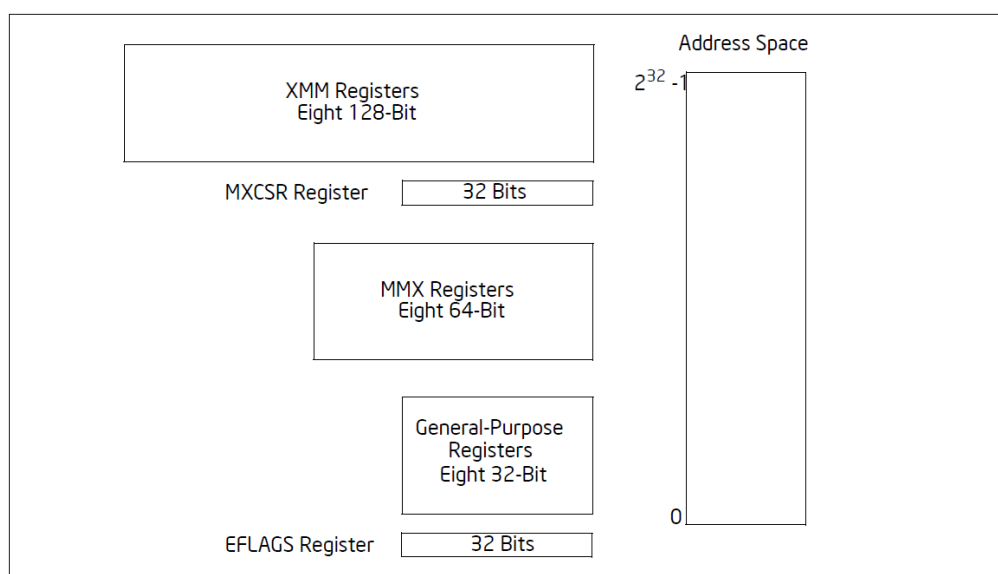


图 4: SSE Execution Environment

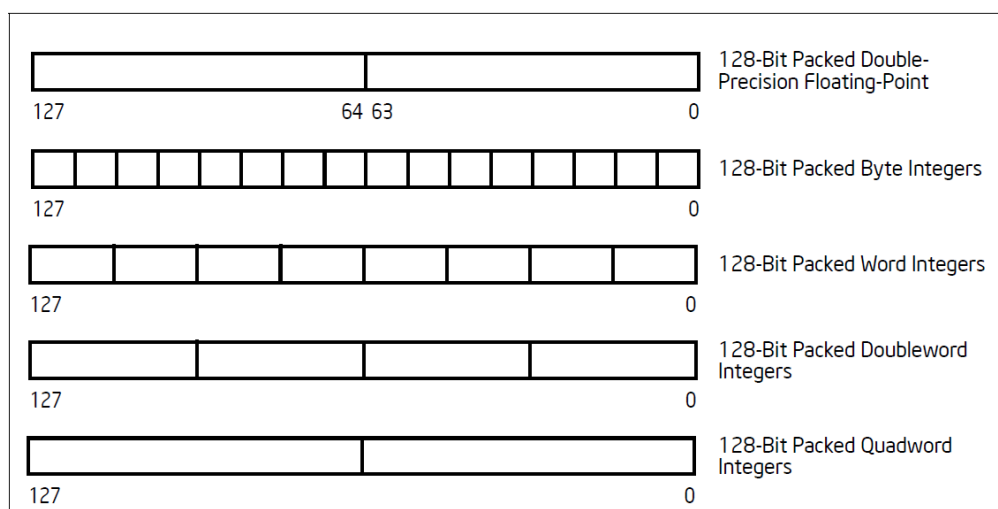


图 5: Data Types Introduced with the SSE2 Extensions

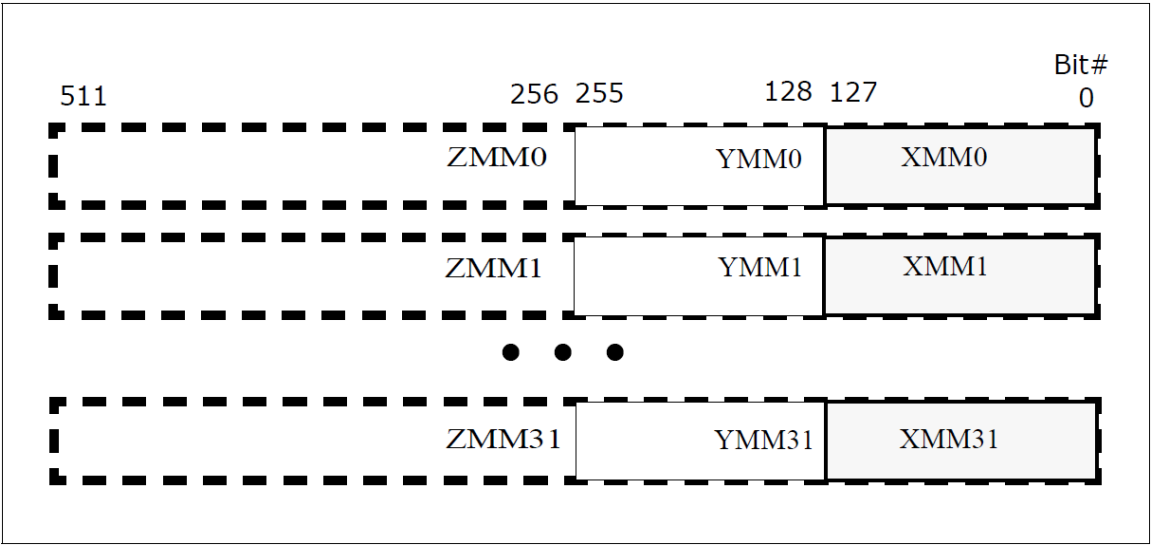


图 6: AVX-512 ZMM

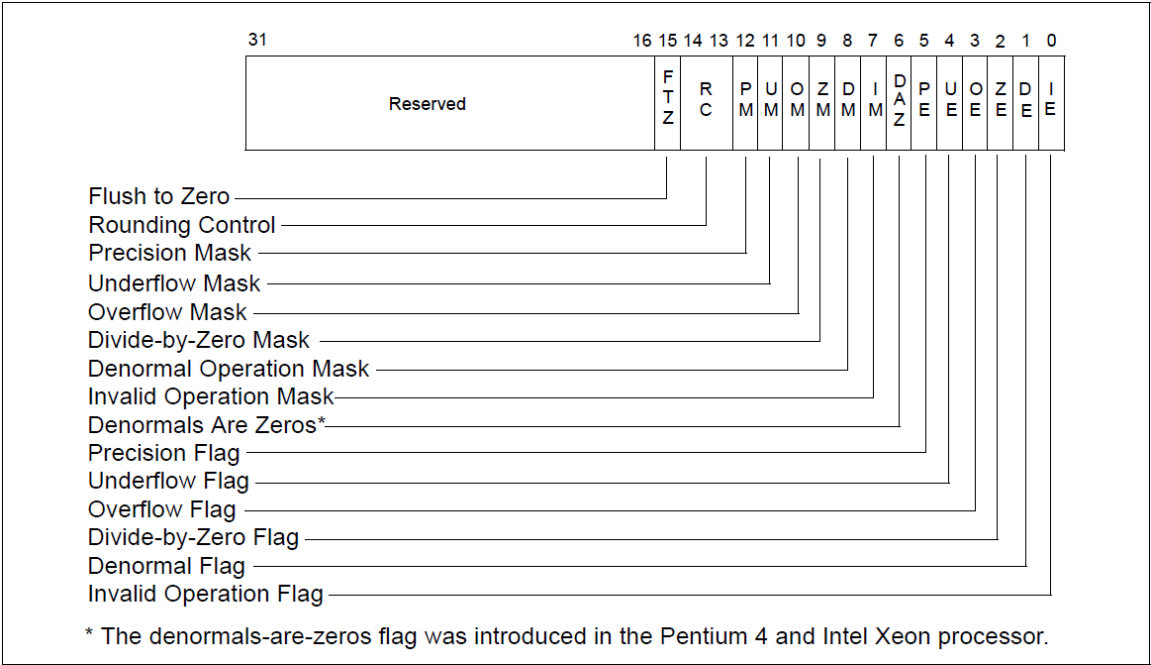


图 7: MXCSR

General-Purpose Registers					
31	16	15	8	7	0
		AH		AL	AX EAX
		BH		BL	BX EBX
		CH		CL	CX ECX
		DH		DL	DX EDX
		BP			EBP
		SI			ESI
		DI			EDI
		SP			ESP

图 8: General-Purpose Register

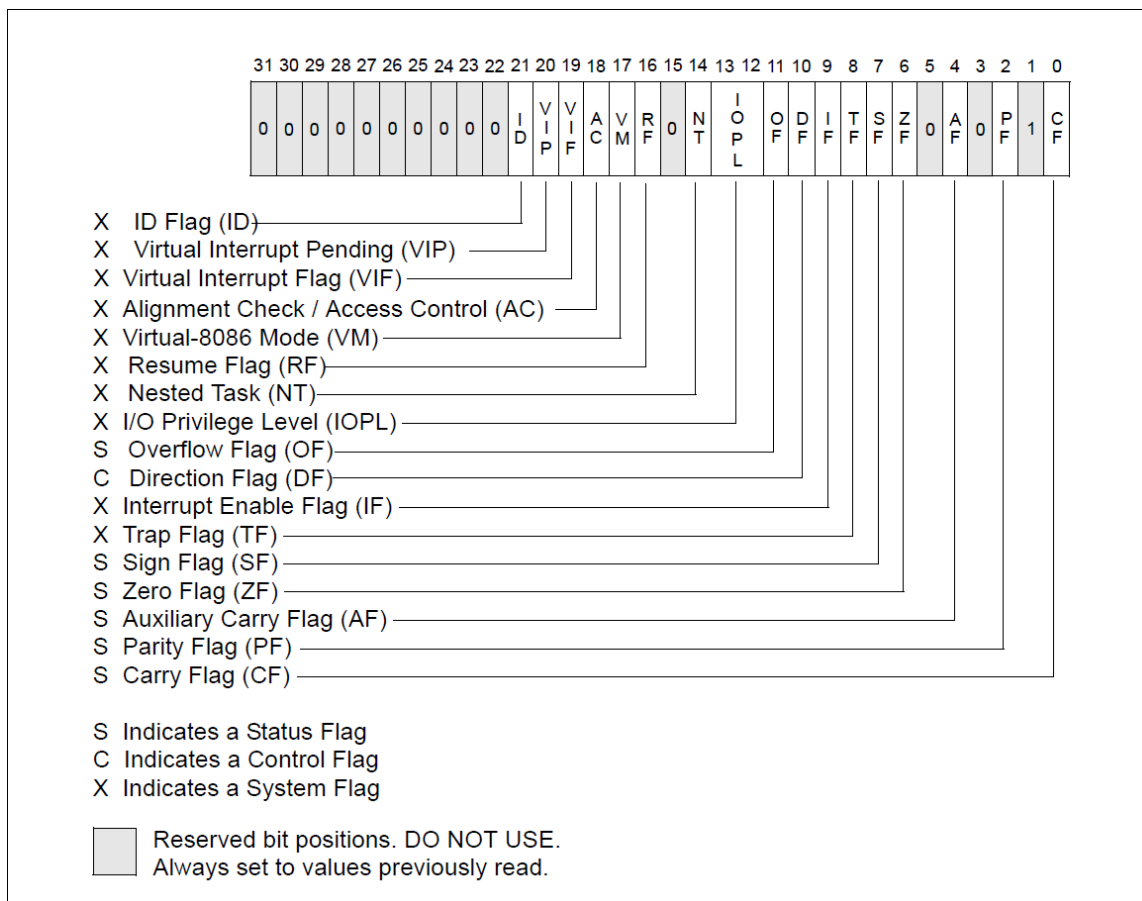


图 9: EFLAG

3 Intrinsics Guide 测试

3.1 测试函数

```

1 #include <immintrin.h>
2 /* 32个int8_t并行相加 */
3 void simd_add_int8_32(int8_t* a, int8_t* b, int8_t* r)
4 {
5     __m256i mma, mmb, mmr;
6     /* load */
7     mma = _mm256_load_si256((__m256i*)a);    // vmovdqa ymm, m256
8     mmb = _mm256_load_si256((__m256i*)b);
9     /* add */
10    mmr = _mm256_add_epi8(mma, mmb);          // vpaddb ymm, ymm, ymm
11    /* store */
12    _mm256_store_si256((__m256i*)r, mmr);    // vmovdqa m256, ymm
13 }
14 /* 8个float并行相加 */
15 void simd_add_float_8(float* a, float* b, float* r)
16 {
17     __m256 mma, mmb, mmr;
18     /* load */
19     mma = _mm256_load_ps(a);                  // vmovaps ymm, m256

```

```

20     mmb = _mm256_load_ps(b);
21     /* add */
22     mmr = _mm256_add_ps(mma, mmb);           // vaddps ymm, ymm, ymm
23     /* store */
24     _mm256_store_ps(r, mmr);                 // vmovaps m256, ymm
25 }

```

3.2 耗时测试

表 1: 3.2E7 次加法运算耗时 (VS compiler)

数据类型	SIMD	非 SIMD
int8_t	3.367s	8.496s
float	12.341s	7.676s

表 2: 3.2E7 次加法运算耗时 (Intel C++ compiler)

数据类型	SIMD	非 SIMD
int8_t	3.045s	8.081s
float	12.272s	7.409s