Objetivo

O objetivo deste documento é apenas explicar um pouco mais dos tipos de dados e funções usadas tanto na codificação de programas AVX <immintrin.h>, quanto NEON <arm_neon.h>.

Tabela 1 - Instruções AVX

Nome	Descrição
__m256	Representa um vetor AVX de 256 bits, que pode armazenar e processar 8 elementos de ponto flutuante de precisão simples com 32 bits cada.
_mm256_set1_ps	Função que inicializa um vetor de 256 bits com valores escalares de ponto flutuante de precisão simples (valores float32) conforme especificado pelo parâmetro.
_mm256_mul_ps (a, b)	Realiza a multiplicação elemento a elemento entre os vetores ‘a’ e ‘b’
_mm256_loadu_ps	Carrega o vetor float para o m256
_mm256_add_ps (a, b)	Soma elemento por elemento dos vetores ‘a’ e ‘b’ e armazena em vetor ‘c’
_mm256_storeu_ps (a, b)	Armazena o vetor ‘b’ no vetor ‘a’.

Tabela 2 - Instruções NEON

Nome	Descrição
float32x4_t	Registra um SIMD de 128 bits, que pode armazenar e processar 4 elementos de ponto flutuante de precisão simples com 32 bits cada.

vdupq_n_f32	Inicializa um vetor com valores escalares de ponto flutuante de precisão simples (valores float32) conforme especificado pelo parâmetro.
vmulq_f32	Realiza a multiplicação dos elementos de ponto flutuante de precisão simples em dois vetores de 128 bits.
vld1q_f32	Carrega um vetor de 128 bits com valores de ponto flutuante de precisão simples a partir de um endereço de memória especificado.
vaddq_f32 (a, b)	Realiza a adição dos elementos de ponto flutuante com precisão simples de dois vetores ‘a’ e ‘b’ de 128 bits.
vst1q_f32	Armazena um vetor de 128 bits com valores de ponto flutuante de precisão simples a partir de um endereço de memória especificado.