عملیات برداری

- Vector addition: Z = X + Y

```
for(i=0; i<n; i++) z[i]=x[i]+y[i];
```

$$\begin{pmatrix} x_1 + y_1 \\ x_2 + y_2 \\ \dots \\ x_n + y_n \end{pmatrix} = \begin{pmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{pmatrix} + \begin{pmatrix} y_1 \\ y_2 \\ \dots \\ y_n \end{pmatrix}$$

- Vector scaling Y = a * X

```
for(i=0; i<n; i++) y[i]=a*x[i];
```

$$\begin{pmatrix} ax_1 \\ ax_2 \\ \dots \\ ax_n \end{pmatrix} = a \begin{pmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{pmatrix}$$

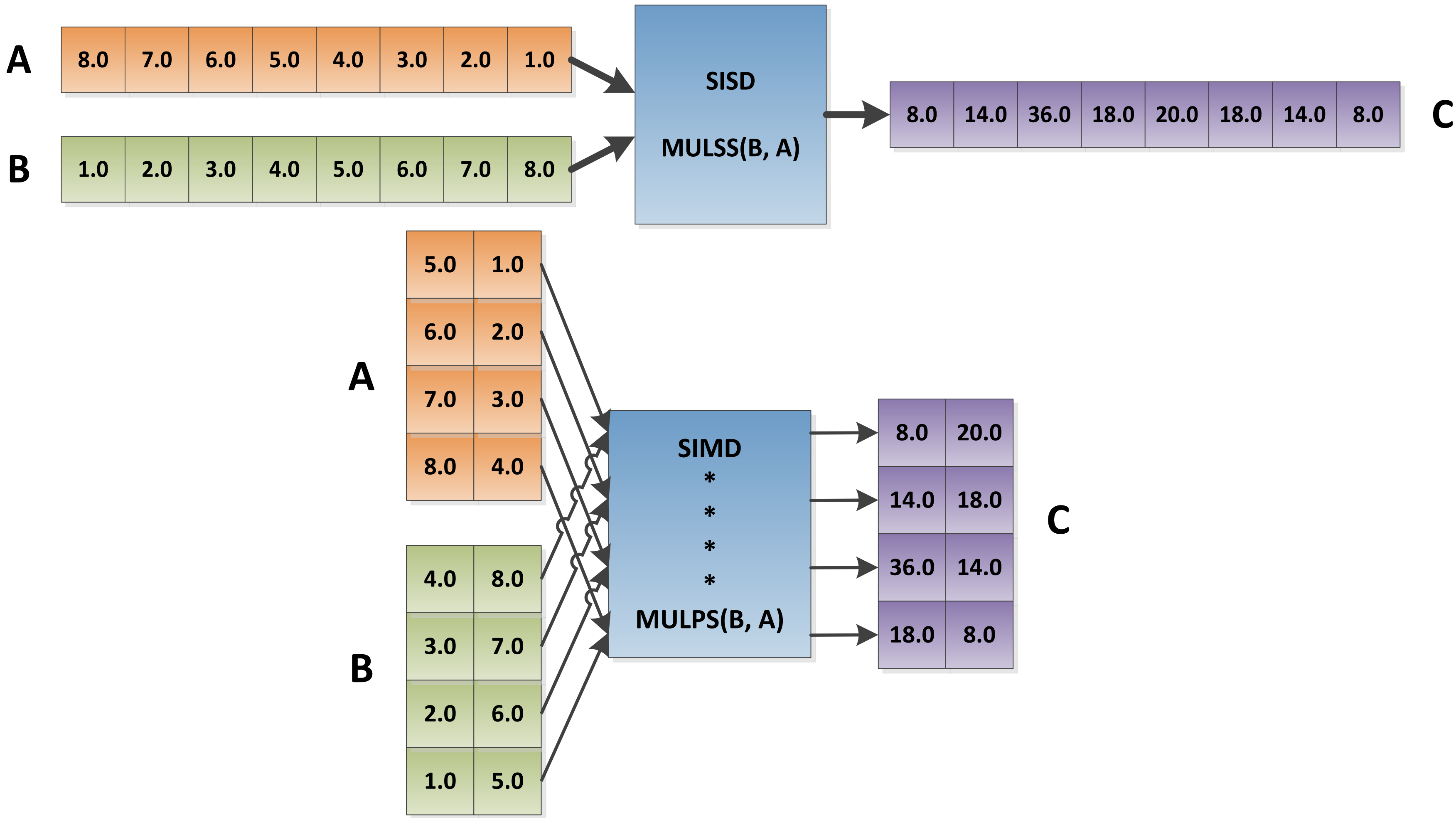
- Dot product Z = X • Y

```
for(i=0; i<n; i++) z+=x[i]*y[i];
```

$$x_1 y_1 + x_2 y_2 + \dots + x_n y_n = \begin{pmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{pmatrix} \bullet \begin{pmatrix} y_1 \\ y_2 \\ \dots \\ y_n \end{pmatrix}$$

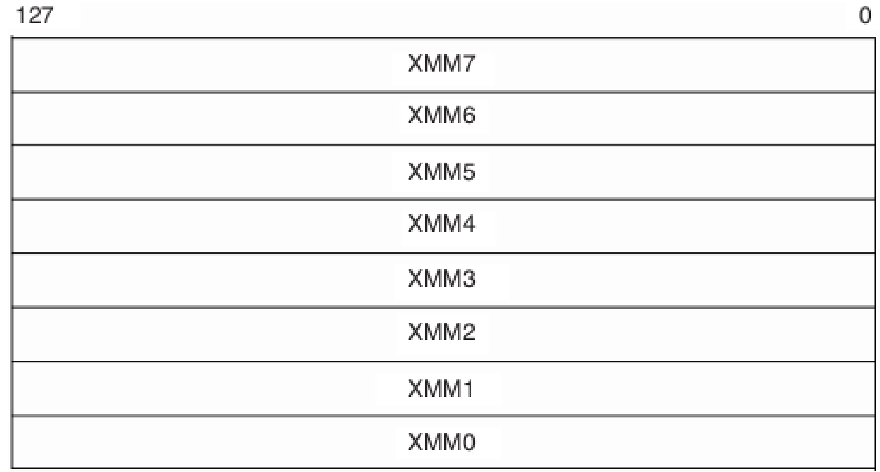
عملیات برداری SISD و SIMD

- C = A * B
  - `for(i=0; i<n; i++) c[i]=a[i]*b[i];`

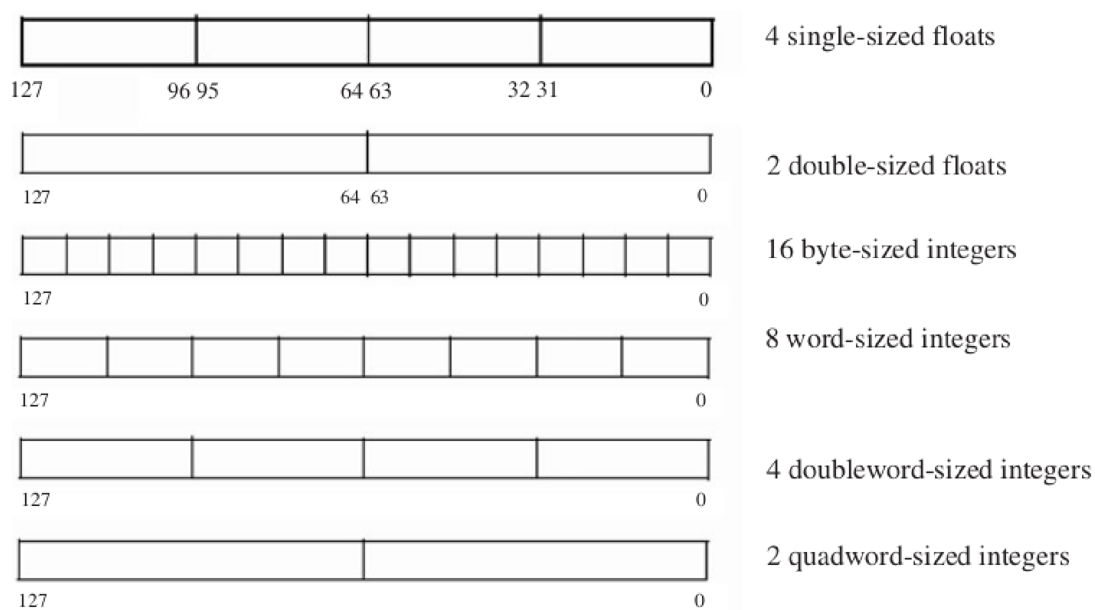


معماری دستورات SSE

The XMM registers used by the SSE instructions.

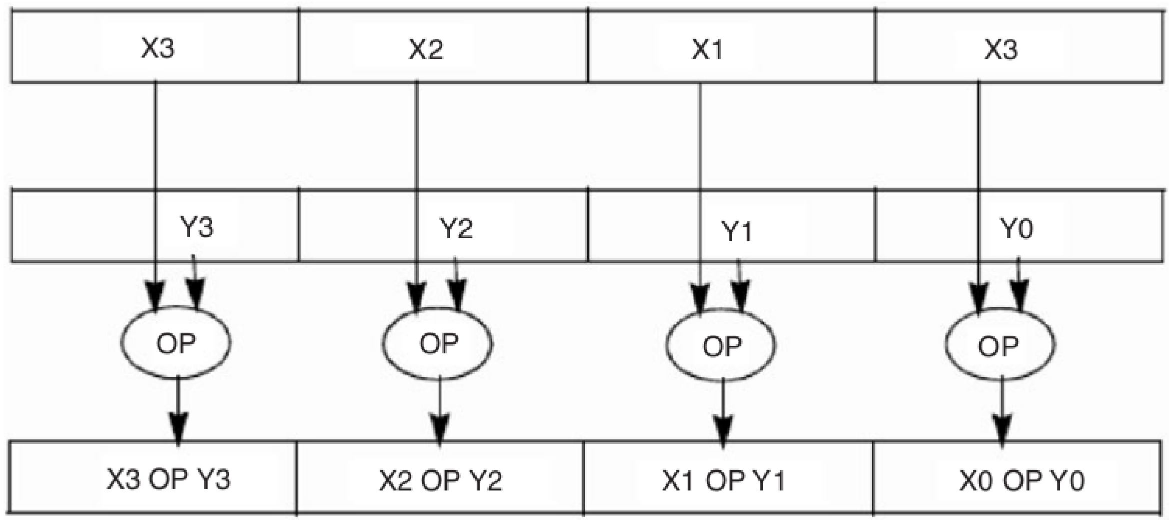


Data formats for the SSE 2 and SSE 3 instructions.

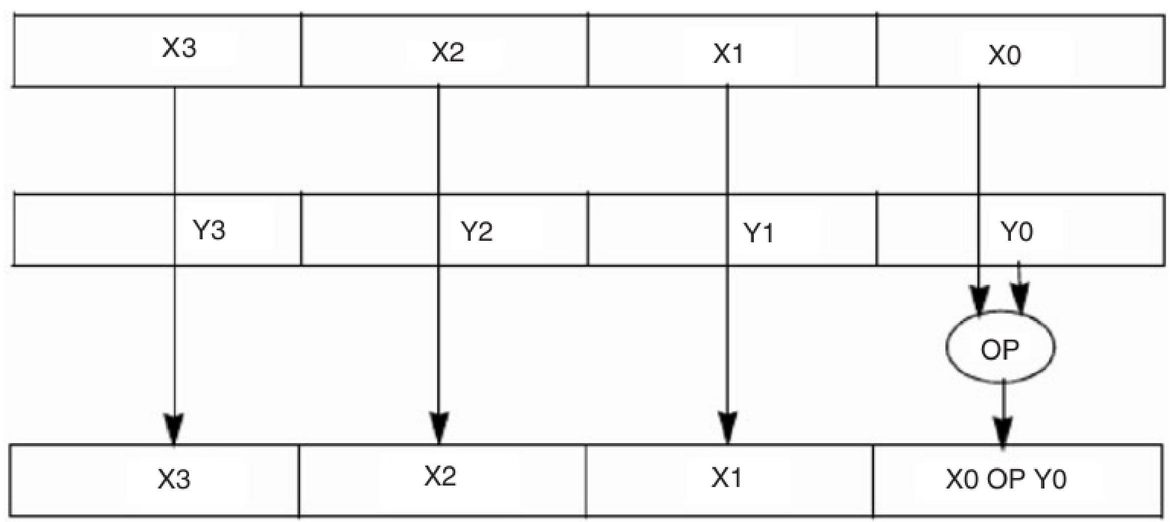


معماری دستورات SSE

Packed single-precision floating point.



Scalar single-precision floating point.



معماری دستورات SSE

Data transfer	Arithmetic	Compare
MOV{A/U}{SS/PS/SD/PD} xmm, mem/xmm	ADD{SS/PS/SD/PD} xmm, mem/xmm	CMP{SS/PS/SD/PD}
	SUB{SS/PS/SD/PD} xmm, mem/xmm	
MOV {H/L} {PS/PD} xmm, mem/xmm	MUL{SS/PS/SD/PD} xmm, mem/xmm	
	DIV{SS/PS/SD/PD} xmm, mem/xmm	
	SQRT{SS/PS/SD/PD} mem/xmm	
	MAX {SS/PS/SD/PD} mem/xmm	
	MIN{SS/PS/SD/PD} mem/xmm	

xmm: one operand is a 128-bit SSE2 register
mem/xmm: other operand is in memory or an SSE2 register
{SS} Scalar Single precision FP: one 32-bit operand in a 128-bit register
{PS} Packed Single precision FP: four 32-bit operands in a 128-bit register
{SD} Scalar Double precision FP: one 64-bit operand in a 128-bit register
{PD} Packed Double precision FP, or two 64-bit operands in a 128-bit register
{A} 128-bit operand is aligned in memory
{U} means the 128-bit operand is unaligned in memory
{H} means move the high half of the 128-bit operand
{L} means move the low half of the 128-bit operand

مراجعه به پیوست B کتاب Brey
جهت مشاهده لیست دستورات

مثال: محاسبه راکتانس مداری با خازن 1.0uF از فرکانس 100Hz تا 10000Hz در گام‌های 100Hz

Align at 16-byte boundaries

$$XC = \frac{1}{2\pi FC}$$

Initialize the pi array with 2*3.1415…

```
void FindXC()
{
	//floating-point example using C++ with the inline assembler

	__declspec(align(16)) float f[4] = {-300,-200,-100,0};
	__declspec(align(16)) float pi[4];
	__declspec(align(16)) float caps[4] = {1.0E-6, 1.0E-6, 1.0E-6, 1.0E-6};
	__declspec(align(16)) float incr[4] = {400, 400, 400, 400};
	__declspec(align(16)) float Xc[400];
	_asm
	{

		fldpi					;form 2π
		fadd	st,st(0)
		fst	pi
		fst	pi+4
		fst	pi+8
		fstp	pi+12
		movaps	xmm0,oword ptr pi
		movaps	xmm1,oword ptr incr
		movaps	xmm3,oword ptr f
		mulps	xmm0,oword ptr caps		;2πC
		mov	ecx,0
LOOP1:
		movaps	xmm2,xmm3
		addps	xmm2,xmm1
		movaps	xmm3,xmm2
		mulps	xmm2,xmm0
		rcpps	xmm2,xmm2			;recipocal
		movaps	oword ptr Xc[ecx],xmm2
		add	ecx,16
		cmp	ecx,400
		jnz	LOOP1
	}
}
```